RICE UNIVERSITY

Discontinuous Galerkin Methods for Parabolic Partial Differential Equations with Random Input Data

by

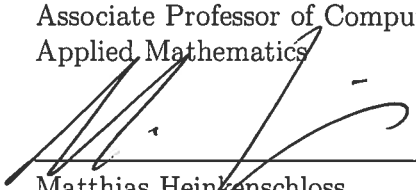
Kun Liu

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

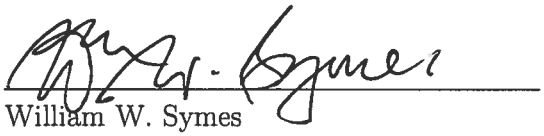
Doctor of Philosophy

Approved, Thesis Committee:

Béatrice M. Rivière, Chair
Associate Professor of Computational and Applied Mathematics



Matthias Heinkenschloss
Professor of Computational and Applied Mathematics



William W. Symes
Noah G. Harding Professor of Computational and Applied Mathematics

Marina Vannucci
Professor of Statistics

Houston, Texas
March, 2013

ABSTRACT

Discontinuous Galerkin Methods for Parabolic Partial Differential Equations with Random Input Data

by

Kun Liu

This thesis discusses and develops one approach to solve parabolic partial differential equations with random input data. The stochastic problem is firstly transformed into a parametrized one by using finite dimensional noise assumption and the truncated Karhunen–Loève expansion. The approach, Monte Carlo discontinuous Galerkin (MCDG) method, randomly generates M realizations of uncertain coefficients and approximates the expected value of the solution by averaging M numerical solutions. This approach is applied to two numerical examples. The first example is a two-dimensional parabolic partial differential equation with random convection term and the second example is a benchmark problem coupling flow and transport equations. I first apply polynomial kernel principal component analysis of second order to generate M realizations of random permeability fields. They are used to obtain M realizations of random convection term computed from solving the flow equation. Using this approach, I solve the transport equation M times corresponding to M velocity realizations. The MCDG solution spreads toward the whole domain from the initial location and the contaminant does not leave the initial location completely as time elapses. The results show that MCDG solution is realistic, because it takes the uncertainty in velocity fields into consideration. Besides, in order to correct overshoot

and undershoot solutions caused by the high level of oscillation in random velocity realizations, I solve the transport equation on meshes of finer resolution than of the permeability, and use a slope limiter as well as lower and upper bound constraints to address this difficulty. Finally, future work is proposed.

Acknowledgements

First of all, I am thankful to my grandma and grandpa for raising me with all their efforts. I am so grateful to my mom and dad for giving me life, love and talents so that I am able to conquer the difficulties and face the challenges with courage.

Most importantly, I want to express my gratitude to my advisor, Professor Béatrice Rivière, who inspires me and teaches me right attitudes in research. Without her guidance, it is impossible for me to graduate from PhD program in CAAM department at Rice University.

I am also very happy to acknowledge my debt to my committee members: Professor Symes, Professor Heinkenschloss and Professor Vannucci. Thank, Dr. Symes and Dr. Heinkenschloss, for their valuable suggestions in CAAM600, through which I improve my presentation skills and get better understanding of the importance of my PhD research. I am also very grateful to Professor Vannucci, who spends her precious time on reviewing my thesis and giving me advice in general.

I also want to thank Daria, Fran, Brenda, Ivy and Jennifer for their excellent assistance and organization in CAAM department. I really appreciate it.

I would like to express my appreciation to Xin Wang and Drew Kouri. Thank them for discussing with me about my research.

Finally, I want to thank God for bringing so many lovely brothers and sisters to Rice Chinese Christian Fellowship. It is unforgettable to live with and rely on them when I am weak. I especially thank my girlfriend Margaret for her support and understanding when I am in difficulty.

Contents

Chapter 1

Introduction

Partial differential equations (PDEs) with random input data is one of the most powerful tools to model oil and gas production as well as groundwater pollution control. However, in most cases, the information available on the input data is very limited, which causes high level of uncertainty in approximating the solution to these problems. The input data in these cases usually depends on the material properties and only part of them are known to certain accuracy. The lack of statistical information on material properties influences the accuracy of solutions to PDEs with random coefficients. For example, it is hard to predict the fluid flow through an oil reservoir, because the permeability of the media is randomly distributed in earth with certain correlation structures. The permeability is measured by making samples at wells, whose volume represents a tiny proportion of the whole reservoir. Therefore, significant uncertainty exists in fluid flow predictions, on which large amount of financial resources is invested. Due to the lack of information to describe the material properties, the development of effective algorithms to solve PDEs with random input data is limited. However, thanks to the development of parallel machines in recent years, powerful techniques are developed and applied to identify the uncertain properties of materials. In my thesis, I apply Kernel Principal Component Analysis (PCA) to identify the random coefficients and develop an effective method to approximate the solutions to parabolic partial differential equations with random input data in a parametrized form. I apply this method to a coupled flow and transfer problem with

uncertain input data. This Ph.D. dissertation extends my Master's thesis [1] which focuses on the numerical solution to elliptic partial differential equations with random coefficients.

1.1 Motivation

Single-phase flow model in porous media with uncertain coefficients is a fundamental model in oil reservoir simulation and is widely used to describe how uncertain factors, such as permeability, impact the pressure of the fluid. In the oil industry, engineers usually employ finite volume method to approximate the solution to the single-phase flow model. However, due to piecewise constant approximation, the solution by finite volume method is of low accuracy. In addition, the uncertainty in this model is not taken into consideration in most flow simulations both in the oil industry and in academic research. These two factors can cause major problems in reservoir management. In this thesis, I apply the theoretical and numerical results obtained in my Master's thesis to the single-phase flow model. By combining a random sampling technique and a locally mass conservative method, the uncertainty of coefficient is sufficiently addressed and high order of accuracy is achieved.

In single-phase flow model, the uncertainty of coefficients comes from the randomness of permeability field in porous media. Although the knowledge of permeability is desperately needed in the oil industry, it is hard to accurately measure the permeability field in earth, due to the large area of oil reservoir and complicated earth structure. A popular method to identify random permeability field is the Source Point Method (SPM) by Ghori, Heller and Singh [2] in 1992. However, the realizations of random permeability by this method are of low variance. Therefore, it is not validated to represent the uncertainty and complexity of random permeability field in earth. An-

other popular method to generate random permeability field is Monte Carlo Markov Chain (MCMC). However, it was proved by Luo and Efendiev [3] in 2006 that MCMC is extremely CPU demanding when applied to generate random permeability fields. Motivated by the dissatisfying properties of these methods, I apply kernel Principle Component Analysis (PCA) to produce realizations of random permeability fields. Kernel PCA is not only computationally less expensive, but also does a better job than K–L expansion to capture more complex geological structures, such as channel. Therefore, the uncertainty of random permeability field is well identified and the solution to single-phase flow model is accurate.

Extended from the single-phase flow model, the time-dependent convection-diffusion equation with random coefficients is discussed in my thesis. It is motivated by modelling fluid flow through porous media in petroleum reservoir and groundwater aquifer simulation. In 2009, Proft and Rivière [4] addressed this equation with deterministic and spatially varying diffusion coefficient. However, the convection term in the transport equation in the coupled system is actually a random field. It is because the convection term is a function of the pressure obtained from the flow equation. Since the permeability in flow equation is random, it leads the pressure to be a random field. Therefore, the convection term in the coupled system is a random field. Especially in the convection-dominated transport equation, the high level of uncertainty in convection term brings more difficulty in solving this coupled system with robust solutions. Besides the uncertainty in the convection term, the convection-diffusion problem has been discretized using finite difference or classical continuous finite element methods combined with slope limiting procedures to suppress the non-physical oscillations in numerical simulations. However, these methods do not possess local mass conservation, high order of accuracy and flexibility for general non-conforming

meshes with various degrees of approximation. Discontinuous Galerkin (DG) methods is one of most recently developed methods, which is equipped with all of these properties. Therefore, I use DG to discretize the single-phase flow model and the coupled flow and transport system in my thesis. By using Monte Carlo sampling technique, the uncertainty in convection term is taken into consideration substantially. In practice, the value of the convection term may randomly oscillate with a high level of uncertainty, because it is determined by the pressure in fluid which is influenced by the permeability of the porous media. And the value of permeability significantly varies in oil reservoirs. For example, the permeability value in SPE10 model ranges from $\mathcal{O}(10^{-8})$ millidarcy to $\mathcal{O}(10^{4})$ millidarcy. In my thesis, I use the permeability values in the SPE10 model to generate random permeability fields and apply them on the single-phase flow model and coupled flow and transport system. In order to obtain robust solutions, I apply the slope limiting procedure to suppress the amount of overshoot and undershoot solutions in numerical simulations.

1.2 Literature Review

Thanks to the rapid growth of computational tools, such as large scale computers and parallel simulators, the territory of the applications of PDEs with uncertain coefficients is explosively expanding. Engineers and mathematicians have developed several numerical methods to approximate the solutions to such PDEs, which include stochastic Galerkin method, Monte Carlo finite element method and stochastic collocation methods.

1.2.1 Approximation in Probability Domain

An on-going research about numerical solutions to PDEs with random input data is the stochastic Galerkin method [5]. This method imposes standard Galerkin approximation such as finite element, finite volume [6], to discretize the space domain. For example, Kleiber and Hien [7] discussed Stochastic Finite Element Method in 1992. They introduced the fundamental methodology to solve such PDEs by using finite element methods. However, the classical finite element method is not suitable to solve the transport problem in porous media, because it is not locally mass conservative. In my thesis, since the numerical experiments require local mass conservation, I apply one of the most recently developed method, discontinuous Galerkin (DG) method, for space discretization. The DG method satisfies the property of local mass conservation.

In 2004, Babuška, Tempone and Zouraris [8] employed Galerkin finite element method to approximate the solution to elliptic partial differential equations with random coefficients. This method employs tensor product polynomial or piecewise polynomial approximation in the probability domain. If the number of random variables is small, the tensor product space is attractive, because it allows to choose various polynomial degrees with respect to each random variable. However, when the number of random variable is moderate or large, the dimension of this tensor product space increases exponentially fast with the number of random variables. This phenomena is called *curse of dimensionality* [9]. Due to this reason, Babuška, Tempone and Zouraris [8] in 2004 proposed the Monte Carlo finite element method (MCFEM) which is less computational expensive than the stochastic Galerkin method if the randomness is described by a large amount of random parameters. In that paper, the authors compared the efficiency between these two methods and proved that for

a large or moderate number of random variables MCFEM is preferred. Motivated by their work, in my thesis, I employ Monte Carlo sampling technique in statistical domain, because the random permeability of a porous medium can have a huge variability. A large number of random variables is needed to achieve high accuracy of approximation of permeability. In addition, Monte Carlo sampling technique can naturally lead to a decoupled system, which is easier to parallelize.

Another disadvantage of stochastic Galerkin method is that if the input data depends non-linearly on the random variables or has unbounded second moment, these two cases lead to a fully coupled system which requires efficient and powerful parallel solvers. In order to reduce the computing cost and improve the computational efficiency with sufficient accuracy, stochastic collocation method has been developed rapidly in recent decades.

The essence of stochastic collocation method is an interpolation method. The implementation of this method is straightforward because it approximates the solution on a finite set of interpolation points (so-called collocation points). However, the crucial issue of this method is how to construct the set of collocation points appropriately, because the choice of the collocation points determines the efficiency of this method. On the other hand, as stochastic collocation method also suffers from the curse of dimensionality, i.e. the number of collocation points grows exponentially in the number of random variables, one should consider a sparse tensor product space instead of the full tensor product space to use this method.

The sparse tensor product was first proposed by Smolyak [10] in 1963. Since then, the sparse grid stochastic collocation method has been developed into an important topic in the area of solving PDEs with random input data. Using sparse grids, one can dramatically reduce the number of collocation points while keeping

a high level of accuracy. In 2008, Nobile, Tempone and Webster [11] proposed an isotropic sparse grid stochastic collocation method for solving elliptic partial differential equations with random coefficients and forcing terms. They used a sparse grid to dramatically reduce the curse of dimensionality while demonstrating (sub)-exponential convergence of the error in the asymptotic regime and algebraic convergence of the error in the pre-asymptotic regime with the moderate number of random variables. Following this work, in 2008, Nobile, Tempone and Webster [12] extended the application of sparse grid from isotropic case to anisotropic case. They employed an anisotropic collocation in stochastic space at the zeros of sparse tensor product polynomials by using Clenshaw-Curtis or Gaussian knots. This method preserves the same convergence rates as the isotropic one, while being more efficient for moderately large-dimensional problems as shown in numerical examples. However, when steep gradients or finite discontinuities occur in the stochastic domain, the stochastic sparse grid collocation methods converge very slowly or even fail. To overcome this drawback, Ma and Zabaras [13] proposed an adaptive stochastic collocation method using piecewise multi-linear hierarchical basis functions in 2009. This method can automatically detect the discontinuity in the stochastic domain and adaptively refine the collocation region accordingly. The authors proved that this method significantly reduces the number of collocation points while achieving the same level of accuracy as conventional sparse grid collocation method.

Although the stochastic collocation method possesses these advantageous properties which render them useful to solve single PDEs with random coefficients, it may not be suitable to approximate the solution to a coupled flow and transport system with random input data. As we know, the collocation method is an interpolation method essentially. It samples the realizations of a random field by computing the

interpolant on a set of collocation points, which are known. Because the convection term in the transport equation is a function of the solution to the flow equation, we cannot express the convection term only by using interpolation functions on a set of collocation points. Instead, each realization of random convection term should depend on the corresponding solution to the flow equation, rather than a set of collocation points. Therefore, I do not apply stochastic collocation method in my thesis. Instead, Monte Carlo method is straightforward and does not require the computation of certain interpolant at collocation points. Therefore, I employ Monte Carlo sampling in stochastic domain in my thesis. However, it is worth mentioning that numerical methods to approximate the solutions to PDEs with random coefficients are developed very rapidly nowadays. In the future work, stochastic collocation method might be preferred to solve coupled PDEs with random input data.

Monte Carlo [14] method is popular for its simplicity, concision, natural parallelization and broad applications. The first major application of Monte Carlo method [15] was in nuclear weapon projects by John von Neumann, Stanislaw Ulam and Nicholas Metropolis in the Los Alamos National Laboratory in the 1940s. Since then, Monte Carlo methods have been widely applied into various areas and have become one of the most important tools to investigate the random behaviour in physical and mathematical systems. The reason for such widely disparate applications is that Monte Carlo methods have been shown to be a very useful statistical and numerical sampling technique to approximate the solutions to stochastic problems which are linear, non-linear, or with too complex parameters that the analytic solution does not exist. The essence of Monte Carlo method is to generate a large number of independent and identically distributed (iid) random realizations of the parameter fields. Then we use these realizations to compute the sample average as the final solution. Due

to its powerful advantages and wide applications, I use Monte Carlo methods to address the uncertainty of the input data in the parabolic partial differential equations in my thesis. Actually, in 2004, Babuška, Tempone and Zouraris [8] proposed the Monte Carlo finite element method to approximate the solution to stochastic elliptic partial differential equations. In that algorithm, they used Monte Carlo method to sample iid realizations of the diffusion and load functions in the stochastic domain. Then, standard Galerkin finite element method is applied in the space domain to compute the numerical solution for each realization. By averaging the results over all realizations, the numerical solution to the stochastic partial differential equation is obtained. They concluded that as the number of the realizations of diffusion and load function increases, this method is convergent in the stochastic domain regardless of any choice of the mesh size, and is convergent with second order in space domain in L^2 sense. Historically, because of the slow convergence rate and requirement of relatively large number of realizations in order to achieve an accurate solution, Monte Carlo method is considered to have its own limitations in practice. However, thanks to the ever-increasing computer power and the development of Monte Carlo method itself, in 2011, Mark and Mordechai [16] summarized the significantly state-of-the-art fields where Monte Carlo methods are applied with an irreplaceable role. These areas contain optical tweezers, microbiological exposure assessment, atmospheric environment, time-gated X-ray imagers, magnetic nanoparticles, gas adsorption in carbon pore structures, GPS signal detection in urban interference environment, solar radio astronomy and so forth. In my thesis, the powerful parallel machine STIC in Research Computing Support Group at Rice University is used to do Monte Carlo simulations. A large number of parallel simulations is accomplished for solving large-scale elliptic and parabolic problems.

1.2.2 Approximation in Space Domain

The literature review above focuses on the approaches to quantify the uncertainty in stochastic domain. After the PDE with random input data is transformed into a parametrized one, conventional Galerkin approaches, such as finite element, finite volume, or finite difference method are applied to approximate the solution to the parametrized problem. Nowadays, standard continuous finite element method [7,8,17] is popular due to its efficiency and high-order convergence rate. However, it does not satisfy local mass conservation, which is a crucial property in reservoir simulation and coupled flow and transport problem. Sun and Liu [18] pointed out that violation of the law of local mass conservation in velocity fields could result in "spurious sources and sinks to transport simulations". It leads to overshoot or undershoot of transport solutions.

In recent years, discontinuous Galerkin (DG) methods have been developed rapidly and applied into various areas such as computational fluid dynamics, reservoir simulation and groundwater aquifer simulation, etc. DG method was first proposed in 1973 by Reed and Hill [19]. In the late 1970s, the symmetric interior penalty Galerkin (SIPG) method was introduced by Wheeler [20] and Arnold [21]. In 1999, the nonsymmetric interior penalty Galerkin (NIPG) method was introduced by Rivière, Wheeler, and Girault [22]. In 2004, the incomplete interior penalty Galerkin (IIPG) method was introduced by Dawson, Sun and Wheeler [23]. The attractive advantages of DG methods include: local mass conservation, complex geometrics, high-order accuracy and the use of irregular meshes with hanging nodes.

Compared with the DG methods, finite difference method is not able to handle complex geometrics. The finite volume method is not capable of achieving high-order accuracy and the standard continuous finite element method lacks the ability of local

mass conservation. In 2010, Ganis, Yotov and Zhong [24] applied stochastic mixed finite element method for flow in porous media with multiple rock types. Even compared with mixed finite element methods which possess local mass conservation, DG method is also advantageous. It is because mixed finite element methods demand an expanded form with more computational cost in order to treat full tensor permeability or diffusivity, while interior penalty DG methods treat the full tensor naturally.

In my thesis, I solve coupled flow and transport problems in porous media with random input data. The numerical methods need to possess the property of local mass conservation and high order accuracy is preferred. DG method is a suitable candidate. In 2005, Sun and Wheeler [25] proved the convergence of DG methods for coupled flow and reactive transport problem. In 2008, Epshteyn and Rivière [26] derived the error estimates for a fully discrete scheme using DG in space and backward Euler in time for a miscible displacement model, which is a coupled flow and transport system. Chapter 3 and 4 in book [27] by Rivière discuss the error estimates and discretization form of DG methods applied to purely parabolic problems and parabolic problems with convection. These works build the theoretical foundations and motivate me to apply DG for solving the problems in my thesis.

1.2.3 Generation of Random Permeability

In the single phase flow model and the flow equation in the coupled system, the uncertainty is caused by the randomness of permeability in porous media. Permeability is a measure of the ability of porous media to allow fluids to pass through it. The permeability of porous media is randomly distributed in earth and displays certain correlation structures. Darcy's law implies that permeability is a key parameter connecting the flow velocity and the gradient of pressure. Therefore, identification of

random permeability is crucial for oil production and environmental cleanup. Many techniques have been developed rapidly to generate random permeability fields. Kriging based method and Markov Chain based method are two of them [28]. Both methods are interpolation methods in essence, i.e., they try to find the permeability values at some locations by interpolating the known permeability value at some points of the domain. Kriging based methods are linear least squares methods. In Chapter 3 of the book [29] in 1999, Chilés and Delfiner discussed this method in detail. Not only they established the formulation of this method, but also analysed the measurement errors. Although this method is linear and easy to be understood, it is expensive, because we need to solve a linear system at each point where permeability is estimated. Because the permeability is estimated directly from the transition matrix, Markov chain based methods are less expensive than Kriging based methods. Furthermore, transition probability is easier to interpret than the variograms in Kriging based methods. Because of these two reasons, Markov Chain based methods are more widely used to model the random permeability field nowadays. In 2007, Li [30] proposed a fixed-path spatial Markov chain algorithm for conditional simulation of discrete spatial variables. Extending Li's work, in 2010, Zein, Rath and Clauser [28] presented multidimensional Markov chain Monte Carlo (MCMC) model to generate the random permeability field. The numerical experiment in that paper showed that multidimensional MCMC is able to identify the permeability field with binary values accurately. However, the accuracy of this method depends on the large number of wells where pressure is known. In practice, wells are located sparsely in the area of oil reservoir. Because of the expensive financial cost of each well, a large number of sample wells is not realistic. Besides, the values of permeability of porous media vary significantly in oil reservoirs. In the numerical experiment in [28], the authors only

used binary value permeability as the target to simulate. The results are not convincing if applied to real reservoir simulation, because the number of permeability values in reality is far more than two and the permeability is of high randomness. Therefore, I do not use MCMC in my thesis to generate random permeability fields. Instead, I apply another state-of-art technique, linear and kernel Principal Component Analysis (PCA), to identify the random permeability of the porous media.

In practice, the permeability of porous media in oil reservoir is a large-scale non-Gaussian, non-stationary random field. Karhunen–Loève expansion is popular to generate a random field, due to its mean-square convergence. However, it is limited by two factors. First, it requires the eigen-decomposition of the covariance matrix of the random field, which can be extremely expensive for large models. Second, it only preserves two-point statistics (mean and covariance). Therefore, it does a poor job to capture complex geological structures, such as channels. In 1998, Schölkopf, Smola and Müller [31] proposed a nonlinear form of PCA. It was proved to be an efficient algorithm in high dimensional feature space, because the eigen-decomposition only performed on the kernel matrix, which is much smaller than the full covariance matrix of a random field. Due to the use of high order polynomial kernels, this method performs very well to reproduce the channel structure in the numerical experiments in that paper. In 2011, Ma and Zabaras [32] applied kernel PCA to reconstruct the channel structure of a large-scale test sample. They concluded that kernel PCA is more effective than linear PCA if the data is nonlinearly related. In my thesis, I generate the random permeability field based on the SPE10 model [33], which is a benchmark problem proposed by the Society of Petroleum Engineers (SPE). In that model, the porous medium is divided into horizontal layers. In the bottom layers, channels appear in the structure of the rock. Therefore, I use second-order polynomial

kernel PCA to reproduce the channel structure in realizations.

In summary, I apply Monte Carlo discontinuous Galerkin method to approximate the solution to elliptic and parabolic partial differential equations with random input data. Applications include single-phase flow model and coupled flow and transport problems.

1.3 Outline

In Chapter 2, I present the formulation of Monte Carlo discontinuous Galerkin (MCDG) methods and apply it to solve a parabolic partial differential equation with random coefficients. Two numerical examples are given. The first example is a one-dimension problem with random diffusion coefficient, random convection term and random load function. The second example is a two-dimensional problem with a random convection term. I propose two approaches to solve this two dimensional problem and compare the simulations.

In Chapter 3, I define the linear and kernel PCA and apply it to generate random permeability fields based on input data from a benchmark problem. Realizations of random permeability fields are then used to approximate the pressure in a single phase flow model.

In Chapter 4, I discuss a coupled flow and transport system with random input data in both equations. One approach is developed to approximate the solution to this coupled system. In the numerical example, I create a benchmark problem and apply this approach to solve it. Finally, I compare the results in two types of heterogeneous porous media.

In Chapter 5, I summarize the conclusions in this thesis and propose future works.

Chapter 2

Model Parabolic Problem

2.1 Introduction

In this chapter, I first transform the parabolic partial differential equations with random input data into parametrized ones by the Karhunen–Loève expansion using a finite dimensional noise assumption. Then, I establish the formulations of Monte Carlo Discontinuous Galerkin (MCDG) method to solve this parametrized problem. The MCDG method is applied to a one dimensional numerical example and a two dimensional example.

2.2 Model

Let $(\Omega, \mathcal{F}, P)$ be a complete probability space, where Ω is the outcome space, $\mathcal{F}$ is the $\sigma-$algebra of events, and $P : \mathcal{F} \to [0, 1]$ is a probability measure. Assume $D \subset \mathbb{R}^d$ is a convex bounded domain. Let $\mathbf{b}(\omega, \cdot)$ be a divergence free vector, i.e. almost surely (a.s.), the following equation holds:

$$\nabla \cdot \mathbf{b}(\omega, \cdot) = 0,$$

where ω is a random variable. The boundary of the space domain ∂D is decomposed into two parts: Dirichlet boundary Λ_D and Neumann boundary Λ_N. Consider a problem of the transport of a chemical species through a porous medium, which is modelled by the following linear parabolic partial differential equations with random

coefficients:

$$\begin{aligned}\frac{\partial u(\omega,\cdot)}{\partial t} + \nabla\cdot(\mathbf{b}(\omega,\cdot)u(\omega,\cdot) - (a(\omega,\cdot)\nabla u(\omega,\cdot))) \\ +c(\omega,\cdot)u(\omega,\cdot) = f(\omega,\cdot) \quad \text{in} \quad (0,T]\times D, \\ u(\omega,\cdot) = u_D(\omega,\cdot) \quad \text{on} \quad (0,T]\times\Lambda_D, \\ -(a(\omega,\cdot)\nabla u(\omega,\cdot))\cdot\mathbf{n} = u_N(\omega,\cdot) \quad \text{on} \quad (0,T]\times\Lambda_N, \\ u(\omega,\cdot) = u_0(w,\cdot) \quad \text{on} \quad \{0\}\times D.\end{aligned} \tag{2.1}$$

where $a, \mathbf{b}, c, f : \Omega \times D \to \mathbb{R}$ are random fields with continuous and bounded covariance functions.

Assume that a is bounded and uniformly coercive, i.e.,

$$\exists a_{min}, a_{max} \in (0,\infty): \quad P(\omega\in\Omega : a(\omega,x)\in[a_{min},a_{max}] \quad a.e.\ x\in\bar{D}) = 1. \tag{2.2}$$

The solution $u : [0,T]\times\Omega\times\bar{D}\to\mathbb{R}$ is a random field that satisfies equations (2.1) $P-$a.e..

The regularity of the solution u can be ensured by the assumption that the first derivative of a is uniformly bounded and continuous, i.e., there exists a constant C such that

$$P(\omega\in\Omega : a(\omega,\cdot)\in C^1(\bar{D}) \quad \text{and} \quad \max_{\bar{D}}|\nabla_x a(\omega,\cdot)| < C) = 1. \tag{2.3}$$

Additionally, the load function f is assumed to satisfy

$$\int_\Omega\int_D f^2(\omega,x)dxdP(\Omega) < \infty, \quad \text{which implies} \quad \int_D f^2(\omega,x)dx < \infty \quad a.s.. \tag{2.4}$$

2.2.1 Karhunen–Loève Expansion

The K–L expansion is an appropriate method to approximate stochastic processes [5]. In this section, I give a detailed description of K–L expansion. Define the integral

operator Q_K for $K : D \times D \to \mathbb{R}^d$:

$$(Q_K f)(x) = \int_D K(x,y) f(y) dy \quad \forall f \in L^2(D) \quad \forall x \in D, \tag{2.5}$$

where D is a convex and bounded domain in $\mathbb{R}^d$.

Theorem 2.1 (Mercer's Theorem [34])

Let K be a real-valued symmetric, continuous function of x and y. Assume in addition that the operator Q_K is non-negative in the usual sense:

$$(Q_K f, f)_{L^2(D)} \geq 0 \quad \forall f \in L^2(D).$$

Then the kernel K can be expanded in a uniformly convergent series

$$K(x,y) = \sum_{i=1}^{\infty} \lambda_i v_i(x) v_i(y),$$

where λ_i and v_i are the non-negative eigenvalues and orthonormal eigenfunctions of Q_K respectively.

By Mercer's Theorem, we obtain the following important result which is known as Karhunen–Loève Theorem.

Theorem 2.2 (Karhunen–Loève Theorem [35])

Consider a centered stochastic process $a(\omega, x)$ with continuous covariance function $cov[a]$. Then the corresponding integral operator $Q_a : L^2(D) \to L^2(D)$ defined by

$$(Q_a f)(\cdot) = \int_D cov[a](x, \cdot) f(x) dx \quad \forall f \in L^2(D), \tag{2.6}$$

has an orthonormal basis of eigenfunctions $\{v_i\}_i$ in $L^2(D)$. If $\phi_i(\omega) = \frac{1}{\sqrt{\lambda_i}} \int_D a(\omega, x) v_i(x) dx$, then ϕ_i are centered random variables that are orthogonal, namely $\mathbb{E}[\phi_i \phi_j] = 0, i \neq j$ and

$$a(\omega, x) = \sum_{i=1}^{\infty} \sqrt{\lambda_i} v_i(x) \phi_i(\omega).$$

Let $\{v_i\}_{i=1}^{\infty}$ be the sequence of the corresponding orthonormal eigenfunctions, i.e.,

$$(v_i, v_j)_{L^2(D)} = \delta_{ij}, i, j \in \mathbb{N}_+.$$

By Mercer's Theorem, we know that the eigenvalues $\{\lambda_i\}_{i=1}^{\infty}$ of the operator Q_a are non-negative. We order these eigenvalues so that the eigenvalues of Q_a are decreasing, i.e., $\lambda_i \geq \lambda_{i+1}, i \in \mathbb{N}_+$.

Proposition 1 Let $\{\lambda_i\}_{i=1}^{\infty}$ be the set of eigenvalues of Q_a. Then

$$\lambda_i \leq \left(\int_{D\times D} (cov[a])^2(x_1, x_2)dx_1dx_2 \right)^{\frac{1}{2}}, \quad \forall i \in \mathbb{N}_+.$$

Proof:

We have

$$\begin{aligned} (Q_a v_i)(x_2) &= \int_D cov[a](x_1, x_2)v_i(x_1)dx_1, \\ (Q_a v_i)(x_2) &= \lambda_i v_i(x_2). \end{aligned}$$

By Cauchy-Schwarz's inequality, this implies

$$\lambda_i \|v_i\|_{L^2(D)} \leq \left(\int_{D\times D} (cov[a])^2(x_1, x_2)dx_1dx_2 \right)^{\frac{1}{2}} \|v_i\|_{L^2(D)}.$$

Since, $\|\phi_i\|_{L^2(D)} = 1$, we obtain

$$\lambda_i \leq \left(\int_{D\times D} (cov[a])^2(x_1, x_2)dx_1dx_2 \right)^{\frac{1}{2}}, \quad \forall i \in \mathbb{N}_+.$$

This is the conclusion we desire. □

Note that $a(\omega, x) - \mathbb{E}[a(\cdot, x)]$ is a centered stochastic process. We apply the Karhunen–Loève theorem to this new stochastic process. Define

$$Y_i(\omega) = \frac{1}{\sqrt{\lambda_i}} \int_D (a(\omega, x) - \mathbb{E}[a(\cdot, x)])v_i(x)dx, \quad \text{for } \lambda_i > 0.$$

Therefore, $Y_i's$ have the properties

$$\mathbb{E}[Y_i] = 0, \tag{2.7}$$

$$\mathbb{E}[Y_i Y_j] = 0, \quad i \neq j, \tag{2.8}$$

$$\mathbb{E}[Y_i^2] = 1. \tag{2.9}$$

By Karhunen–Loève Theorem, the stochastic functions $a(\omega, x), \mathbf{b}(\omega, x), c(\omega, x), f(\omega, x)$ can be written as

$$a(\omega, x) = \mathbb{E}[a(\cdot, x)] + \sum_{i=1}^{\infty} \sqrt{\lambda_i} v_i(x) Y_i(\omega), \tag{2.10}$$

$$\mathbf{b}(\omega, x) = \mathbb{E}[\mathbf{b}(\cdot, x)] + \sum_{i=1}^{\infty} \sqrt{\bar{\lambda}_i} \bar{v}_i(x) \bar{\mathbf{Y}}_i(\omega), \tag{2.11}$$

$$c(\omega, x) = \mathbb{E}[c(\cdot, x)] + \sum_{i=1}^{\infty} \sqrt{\check{\lambda}_i} \check{v}_i(x) \check{Y}_i(\omega), \tag{2.12}$$

$$f(\omega, x) = \mathbb{E}[f(\cdot, x)] + \sum_{i=1}^{\infty} \sqrt{\hat{\lambda}_i} \hat{v}_i(x) \hat{Y}_i(\omega). \tag{2.13}$$

Then we can approximate the stochastic functions $a(\omega, x), \mathbf{b}(\omega, x), c(\omega, x), f(\omega, x)$ by the truncated Karhunen–Loève expansions $a_N(\omega, x), \mathbf{b}_N(\omega, x), c_N(\omega, x), f_N(\omega, x)$ which are defined by

$$a_N(\omega, x) = \mathbb{E}[a(\cdot, x)] + \sum_{i=1}^{N} \sqrt{\lambda_i} v_i(x) Y_i(\omega) \quad N \in \mathbb{N}_+, \tag{2.14}$$

$$\mathbf{b}_N(\omega, x) = \mathbb{E}[\mathbf{b}(\cdot, x)] + \sum_{i=1}^{N} \sqrt{\bar{\lambda}_i} \bar{v}_i(x) \bar{\mathbf{Y}}_i(\omega) \quad N \in \mathbb{N}_+, \tag{2.15}$$

$$c_N(\omega, x) = \mathbb{E}[c(\cdot, x)] + \sum_{i=1}^{N} \sqrt{\check{\lambda}_i} \check{v}_i(x) \check{Y}_i(\omega) \quad N \in \mathbb{N}_+, \tag{2.16}$$

$$f_N(\omega, x) = \mathbb{E}[f(\cdot, x)] + \sum_{i=1}^{N} \sqrt{\hat{\lambda}_i} \hat{v}_i(x) \hat{Y}_i(\omega) \quad N \in \mathbb{N}_+. \tag{2.17}$$

2.2.2 Finite Dimensional Noise

In this thesis, we assume the diffusion coefficient a, the convection term $\mathbf{b}$, the random field c and the load function f have the form:

$$a(\omega, x) = a(Y_1(\omega), \ldots, Y_N(\omega), x) \quad \text{on} \quad \Omega \times \bar{D}, \tag{2.18}$$

$$\mathbf{b}(\omega, x) = \mathbf{b}(\mathbf{Y}_1(\omega), \ldots, \mathbf{Y}_N(\omega), x) \quad \text{on} \quad \Omega \times \bar{D}, \tag{2.19}$$

$$c(\omega, x) = c(Y_1(\omega), \ldots, Y_N(\omega), x) \quad \text{on} \quad \Omega \times \bar{D}, \tag{2.20}$$

$$f(\omega, x) = f(Y_1(\omega), \ldots, Y_N(\omega), x) \quad \text{on} \quad \Omega \times \bar{D}, \tag{2.21}$$

and satisfy the truncated finite Karhunen–Loève expansion, i.e.

$$a(\omega, x) = \mathbb{E}[a(\cdot, x)] + \sum_{i=1}^{N} \sqrt{\lambda_i} v_i(x) Y_i(\omega) \quad N \in \mathbb{N}_+, \tag{2.22}$$

$$\mathbf{b}(\omega, x) = \mathbb{E}[\mathbf{b}(\cdot, x)] + \sum_{i=1}^{N} \sqrt{\bar{\lambda}_i} \bar{v}_i(x) \bar{\mathbf{Y}}_i(\omega) \quad N \in \mathbb{N}_+, \tag{2.23}$$

$$c(\omega, x) = \mathbb{E}[b(\cdot, x)] + \sum_{i=1}^{N} \sqrt{\check{\lambda}_i} \check{v}_i(x) \check{Y}_i(\omega) \quad N \in \mathbb{N}_+, \tag{2.24}$$

$$f(\omega, x) = \mathbb{E}[f(\cdot, x)] + \sum_{i=1}^{N} \sqrt{\hat{\lambda}_i} \hat{v}_i(x) \hat{Y}_i(\omega) \quad N \in \mathbb{N}_+. \tag{2.25}$$

where $N \in \mathbb{N}_+, \{Y_i\}_{i=1}^N$ are real-valued random variables with mean value zero and unit variance. Moreover, I assume that for $i = 1, \ldots, N$:

- the image $\Gamma_i = Y_i(\Omega)$ is a bounded interval in $\mathbb{R}$.
- each Y_i has a density function $\rho_i : \Gamma_i \to \mathbb{R}^+$.

In the following, we denote $\rho = \rho(y) : \Gamma \to \mathbb{R}^+$ for the joint probability density of the random vector $Y = (Y_1, \ldots, Y_N)$, where $\Gamma = \prod_{i=1}^N \Gamma_i \subset \mathbb{R}^N$ is the support of the joint probability. We also assume that the random variables $\{Y_i\}_{i=1}^N$ are independent, thus

we have

$$\rho(y) = \prod_{i=1}^{N} \rho_i(y_i) \quad \forall y \in \Gamma.$$

If we apply the Doob-Dynkin lemma [36] with the finite dimensional noise assumption, we know that the solution u to model problem (2.1) can be expressed by a finite number of random variables, i.e. $u(\omega, x) = u(Y_1(\omega), \ldots, Y_N(\omega), x)$. Now, instead of approximating $u(\omega, \cdot)$, our task is to approximate $u(y, x)$, where y belongs to Γ and x belongs to D. Therefore, the stochastic partial differential equations (2.1) are transformed into parabolic partial differential equations with an $N-$dimensional parameter, i.e.,

$$\begin{aligned}
\frac{\partial u(t,y,x)}{\partial t} + \nabla \cdot (\mathbf{b}(y,x)u(t,y,x) - (a(y,x)\nabla u(t,y,x))) \\
+c(y,x)u(t,y,x) = f(t,y,x), \quad t \in (0,T], y \in \Gamma, x \in D \\
u(t,y,x) = u_D(t,y,x), \quad t \in (0,T], y \in \Gamma, x \in \Lambda_D \\
-(a(y,x)\nabla u(t,y,x)) \cdot \mathbf{n} = u_N(t,y,x), \quad t \in (0,T], y \in \Gamma, x \in \Lambda_N \\
u(t,y,x) = u_0, \quad t = 0, y \in \Gamma, x \in D.
\end{aligned} \tag{2.26}$$

Thus, the finite dimensional noise assumption helps us transform the stochastic parabolic problem into a parametrized parabolic problem. At this point, if we sample the random fields $a(y, \cdot), \mathbf{b}(y, \cdot), c(y, \cdot), f(y, \cdot)$, then we can use discontinuous Galerkin methods to approximate the solution to (2.26), as described in the next section.

2.3 Monte Carlo Discontinuous Galerkin Methods

2.3.1 Discontinuous Galerkin Space

Before presenting the definition of discontinuous Galerkin space, we introduce the broken Sobolev spaces, which are the natural spaces for DG methods. Let D be a

polygonal domain which is partitioned into elements E. For two dimensional domains, E is a triangle or a quadrilateral and for three dimensional domains, E is a tetrahedron or a hexahedron. In this thesis, we only consider conforming meshes, i.e. the intersection of two elements in the mesh is either empty, a vertex, an edge, or a face. We now give some notation:

- $\mathcal{E}_h$: the subdivision (or mesh) of the domain D,
- h_E: the diameter of element E,
- h: the maximum element diameter in the mesh,
- κ_E: the maximum diameter of a ball inscribed in element E.

We assume that the subdivision is *regular*, which means there is a constant $\kappa > 0$ such that

$$\frac{h_E}{\kappa_E} \leq \kappa, \quad \forall E \in \mathcal{E}_h.$$

Now, we are ready to introduce the *broken Sobolev spaces*.

Definition 2.1 The *broken Sobolev spaces* for any real number s are defined :

$$H^s(\mathcal{E}_h) = \left\{ u \in L^2(D) : \forall E \in \mathcal{E}_h, u|_E \in H^s(E) \right\},$$

equipped with the broken Sobolev norm:

$$\|u\|_{H^s(\mathcal{E}_h)} = \left(\sum_{E \in \mathcal{E}_h} \|u\|^2_{H^s(E)} \right)^{1/2}.$$

Then, we introduce the DG spaces which are finite dimensional subspaces of the broken Sobolev space $H^s(\mathcal{E}_h)$. Let r be a positive integer. The discontinuous Galerkin spaces are defined as:

$$D_r(\mathcal{E}_h) = \left\{ u \in L^2(D) : \forall E \in \mathcal{E}_h, u|_E \in \mathbb{P}_r(E) \right\}, \tag{2.27}$$

where $\mathbb{P}_r(E)$ denotes the space of polynomials of total degree less than or equal to r on E.

We now define the concepts of the jumps and averages related to the elements in the subdivision $\mathcal{E}_h$ to prepare for the introduction of the DG variational formulation. Denote by D_h the set of interior edges (or faces) of the subdivision $\mathcal{E}_h$. With each edge (or face) e, we associate a unit normal vector $\mathbf{n}_e$. If e is on the boundary ∂D, then $\mathbf{n}_e$ is the outward vector normal to ∂D. If two elements E_1^e and E_2^e are neighbours with one common side e, there are two traces of u along e. We also assume that the normal vector $\mathbf{n}_e$ is oriented from E_1^e to E_2^e, and define the average and jump as:

$$\{u\} = \frac{1}{2}(u|_{E_1^e}) + \frac{1}{2}(u|_{E_2^e}), \quad [u] = (u|_{E_1^e}) - (u|_{E_2^e}) \quad \forall e = \partial E_1^e \cap \partial E_2^e.$$

We extend the definition to sides that belong to the boundary ∂D:

$$\{u\} = [u] = u|_{E_1^e}, \quad \forall e = \partial E_1^e \cap \partial D.$$

2.3.2 Variational Formulation

Problem (2.26) is first discretized in space domain by the DG methods. First, the diffusive term $\nabla \cdot (a\nabla u)$ is discretized by the usual DG bilinear form, $A_\epsilon : H^s(\mathcal{E}_h) \times H^s(\mathcal{E}_h) \to \mathbb{R}$, for $s > \frac{3}{2}$

$$\begin{aligned} A_\epsilon(y; u, v) = &\sum_{E \in \mathcal{E}_h} \int_E a(y, \cdot)\nabla u \cdot \nabla v - \sum_{e \in D_h \cup \Lambda_D} \int_e \{a(y, \cdot)\nabla u \cdot \mathbf{n}_e\}\,[v] \\ &+ \epsilon \sum_{e \in D_h \cup \Lambda_D} \int_e \{a(y, \cdot)\nabla v \cdot \mathbf{n}_e\}\,[u] + \sum_{e \in D_h \cup \Lambda_D} \frac{\sigma}{h} \int_e [v][w], \end{aligned} \tag{2.28}$$

where ϵ belongs to $\{0, +1, -1\}$, σ is the penalty parameter, which is a non-negative real number. The set D_h is the set of interior edges (or faces) of the subdivision $\mathcal{E}_h$.

Remarks:

- If $\epsilon = +1$, the resulting method is called the nonsymmetric interior penalty Galerkin (NIPG) method. Any positive value σ is acceptable. In general, we choose $\sigma = 1$.

- If $\epsilon = -1$ and σ is bounded below by a large enough constant, the resulting method is called the symmetric interior penalty Galerkin (SIPG) method.

- If $\epsilon = 0$ and σ is bounded below by a large enough constant, we obtain the incomplete interior penalty Galerkin (IIPG) method.

Second, the convection term $\nabla \cdot (\mathbf{b}u)$ is approximated by an upwind discretization. Let us denote the upwind value of a function Z by Z^{up}:

$$Z^{up} = \begin{cases} Z|_{E_e^1} & \text{if } \mathbf{b} \cdot \mathbf{n}_e \geq 0 \\ Z|_{E_e^2} & \text{if } \mathbf{b} \cdot \mathbf{n}_e < 0 \end{cases} \qquad \forall e = \partial E_e^1 \cap \partial E_e^2.$$

The dependence of the upwind discretization B on the random variable y is explicitly given:

$$\begin{aligned} B(y; u, v) = -\sum_{E \in \mathcal{E}_h} \int_E (\mathbf{b}(y, \cdot)u) \cdot \nabla v + \sum_{e \in D_h} \int_e \{\mathbf{b}(y, \cdot) \cdot \mathbf{n}_e\} \, u^{up}[v] \\ + \sum_{e \in \Lambda_D \cup \Lambda_N} \int_e \mathbf{b}(y, \cdot) \cdot \mathbf{n}_e uv. \end{aligned} \tag{2.29}$$

The term cu is approximated by:

$$C(y; u, v) = \sum_{E \in \mathcal{E}_h} \int_E c(y, \cdot)uv. \tag{2.30}$$

We choose backward Euler discretization for the time integration. Let Δt denote the time step. The resulting fully discrete variational formulation is: find a sequence

$(u_h^m)_{m\geq 0}$ of functions in $\mathcal{D}_r(\mathcal{E}_h)$ such that: $\forall m \geq 0, \quad \forall v \in \mathcal{D}_r(\mathcal{E}_h)$

$$\begin{aligned}\left(\frac{u_h^{m+1}(y,\cdot)-u_h^m(y,\cdot)}{\Delta t}, v\right)_{L^2(D)} &+ A_\epsilon(y, u_h^{m+1}(y,\cdot), v) + B(y, u_h^{m+1}(y,\cdot), v) \\ &+C(y, u_h^{m+1}(y,\cdot), v) = L(u_h^{m+1}(y,\cdot), v), \\ &(u_h^0, v)_{L^2(D)} = (u_0, v)_{L^2(D)},\end{aligned} \tag{2.31}$$

where y is the realization of the random variable and

$$L(u,v) = \int_D f(y,\cdot)v + \epsilon \sum_{e\in\Lambda_D} \int_e \left(a(y,\cdot)\nabla v \cdot \mathbf{n}_e + \frac{\sigma}{h} v\right) u_D(y,\cdot) + \sum_{e\in\Lambda_N} \int_e v u_N(y,\cdot). \tag{2.32}$$

2.3.3 Monte Carlo Discontinuous Galerkin Methods

Based on section 2.3.1 and section 2.3.2, the formulation of the MCDG method is:

- Choose a number of realizations, $M \in \mathbb{N}_+$, and a DG space $D_r(\mathcal{E}_h)$.

- For each $j = 1, ..., M$, sample iid realizations of $a(y,\cdot), \mathbf{b}(y,\cdot), c(y,\cdot), f(y,\cdot)$ and find $(u_h^m(y_j,\cdot) \in D_r(\mathcal{E}_h))_{m\geq 0}$ such that

$$\begin{aligned}\left(\frac{u_h^{m+1}(y_j,\cdot)-u_h^m(y_j,\cdot)}{\Delta t}, v\right)_{L^2(D)} &+ A_\epsilon(y_i, u_h^{m+1}(y_j,\cdot), v) + B(y_i, u_h^{m+1}(y_j,\cdot), v) \\ &+C(y_i, u_h^{m+1}(y_j,\cdot), v) = L(u_h^{m+1}(y_j,\cdot), v), \quad m = 0, 1, \cdots, N_t - 1, \\ &(u_h^0, v)_{L^2(D)} = (u_0, v)_{L^2(D)},\end{aligned} \tag{2.33}$$

  where N_t is the total number of time steps, A_ϵ is defined in (2.28), B is defined in (2.29), C is defined in (2.30) and L is defined in (2.32).

- Approximate the expected value of the solution at time T by:

$$\mathbb{E}(u_h; M) = \frac{1}{M}\sum_{j=1}^{M} u_h^{N_t}(y_j;\cdot).$$

2.4 Numerical Example

2.4.1 One Dimensional Parabolic Example

In this section, we use MCDG to solve a one dimensional parabolic example with random input data. The model problem is rewritten:

$$\begin{aligned} u_t(t,y,x) - a(y)u''(t,y,x) + b(y)u'(t,y,x) + c(y)u(t,y,x) &= f(y,x), \\ u(t,y,0) &= 1, \\ u(t,y,1) &= cos(\pi t), \\ u(0,y,x) &= 1, \end{aligned}$$

in the time domain $[0,1]$ and space domain $[0,1]$.

The outcome set is $\Gamma = N(0,1)^5$, i.e., five dimensional space with each dimension satisfying the standard normal distribution. The random input functions are:

$$\begin{aligned} a(\mathbf{y}) &= \pi^2 + 10\sum_{i=1}^{5}(\cos(2\pi i y_i^2))^2, \\ b(\mathbf{y}) &= \pi^2 + 8\sum_{i=1}^{5}\sin(2\pi i y_i), \\ c(\mathbf{y}) &= 6\sum_{i=1}^{5}\cos(\pi i y_i)\sin(\pi i y_i), \\ f(\mathbf{y}) &= 10\pi^2\sin(2\pi x t) + 4\sum_{i=1}^{5}\sin(2\pi i y_i)\exp(-i y_i^2), \end{aligned}$$

where $\mathbf{y} = (y_1, \cdots, y_5)$. The NIPG method with penalty parameter $\sigma = 10$ is utilized.

We take 10000 simulations with number of intervals $N = 5, 10, 20, 40$ in space domain and number of time steps $N_t = 2N^2$ in time domain. Quadratic polynomials are used as local basis functions in the DG method. Fig 2.1 shows the MCDG solutions with 1 and 10000 simulations for different numbers of intervals N. The plot in the second row in Fig 2.1 implies MCDG solutions converge when we increase

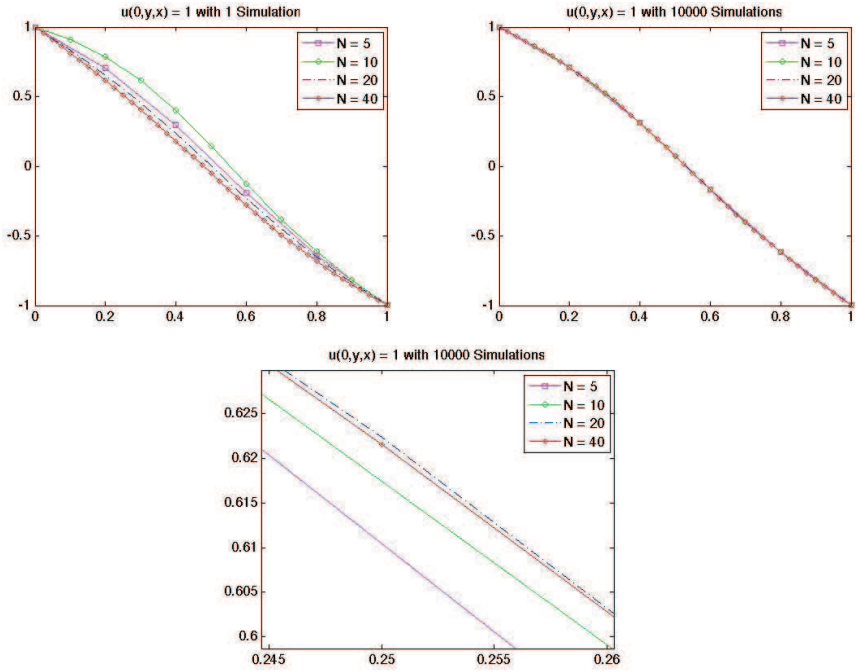


Figure 2.1 : Expected value of 1 (left in the first row) and 10000 (right in the first row) simulations by MCDG method with number of intervals $N = 5, 10, 20, 40$ in space domain. The figure in the second row is the zoom-in observation of MCDG solution by 10000 simulations.

the number of intervals in space domain. The two plots in the first row of Fig 2.1 show that when we take a small number of simulations, the MCDG solutions vary for different numbers of intervals in space domain. When a large number of simulations is taken, the MCDG solutions converge.

2.4.2 Two Dimensional Parabolic Example

In this section, I use two approaches to approximate the solution to a two dimensional parabolic problem, where the convection coefficient is randomly distributed.

The following parabolic equation is considered:

$$\begin{aligned} \frac{\partial u}{\partial t} - \nabla \cdot (a\nabla u - \mathbf{b}u) = 0 \quad &\text{in} \quad (0,T) \times D, \\ u = 0 \quad &\text{on} \quad (0,T) \times \{\Lambda_{D1} \cup \Lambda_{D2}\}, \\ a\nabla u \cdot \mathbf{n} = 0 \quad &\text{on} \quad (0,T) \times \{\Lambda_{N1} \cup \Lambda_{N2}\}, \\ u(0,\mathbf{x}) = u_0, \quad &\text{on} \quad \{0\} \times D, \end{aligned} \tag{2.34}$$

where

$$u_0 = \begin{cases} e^{1/r^2} e^{-1/(r^2-(x-x_0)^2-(y-y_0)^2)} & \text{in} \quad D_0, \\ 0 & \text{elsewhere.} \end{cases}$$

Fig 2.2 shows the initial solution to this parabolic equation. Domain D_0 is a circle with center $(0.2, 0.2)$ and radius $r = 0.2$.

The space domain is $(0,1) \times (0,1)$, which is given in Fig 2.3. The time domain is $[0,1]$ and the diffusion coefficient a is equal to 0.01. I define the convection coefficient $\mathbf{b} = \mathbf{S} + C\boldsymbol{\lambda}$, where $\mathbf{S}$ is a two dimensional constant vector, C is a scalar, $\boldsymbol{\lambda}$ is a two dimensional random vector. Two components of $\boldsymbol{\lambda}$ are independent with each other and satisfy standard normal distribution with mean value 0 and unit variance.

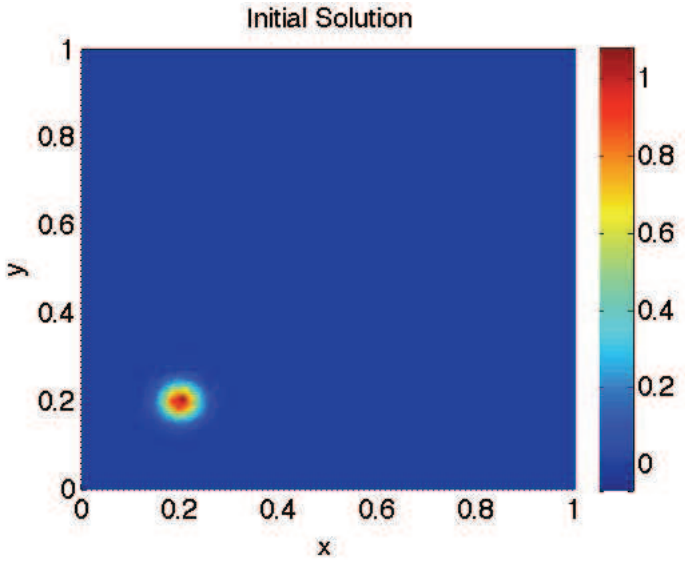


Figure 2.2 : Initial Solution to Parabolic Equation (2.34)

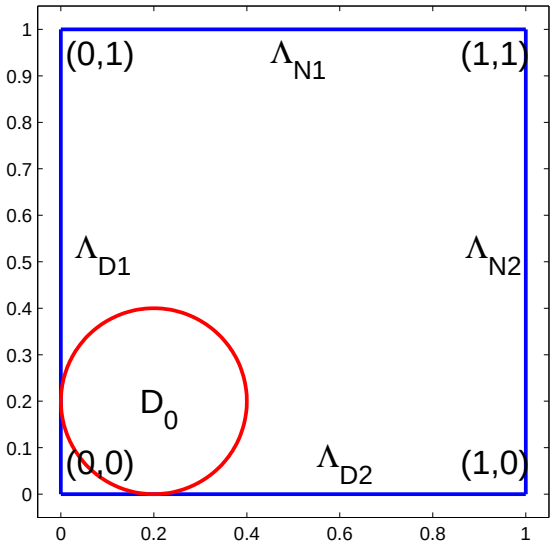


Figure 2.3 : Domain of the two dimensional numerical example with Neumann and Dirichlet boundary conditions. $\Lambda_{D1}, \Lambda_{D2}$ are Dirichlet boundaries. $\Lambda_{N1}, \Lambda_{N2}$ are Neumann boundaries.

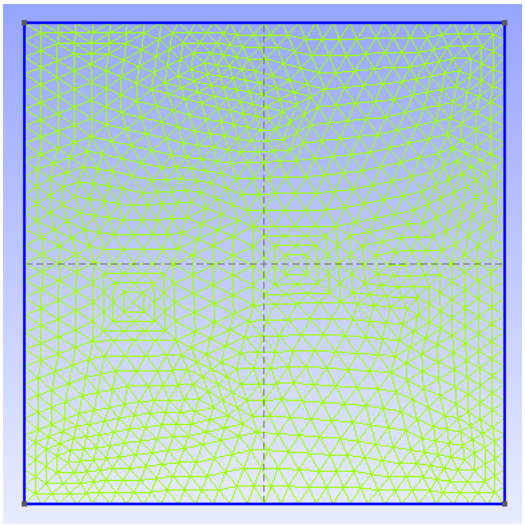


Figure 2.4 : Unstructured mesh with 1033 vertices and 1952 elements.

Approach One: Random Convection Coefficient b

I first apply MCDG on this parabolic problem (2.34) where $\mathbf{S} = (1.0, 1.0)^T$ and $C \in \{0.01, 0.1, 1.0\}$. Therefore, we can observe the influence of different levels of uncertainty of the convection coefficient on the MCDG solutions. I use NIPG with penalty parameter σ equal to 1.0 to discretize this parabolic equation in space domain and use backward Euler method to discretize the time domain. In this example, the unstructured mesh consists of 1033 vertices and 1952 triangular elements. The mesh is generated by Gmsh [37] and is given in Fig 2.4.

I take 12800 simulations using MCDG. Due to the large number of simulations, I use 128 processors of the parallel machine STIC in Rice Research Computing Support Group and allocate 100 simulations on each processor. In Fig 2.5, the first row shows the MCDG solutions when $T = 0.1, 0.2, 0.4$ and $C = 0.01$. The second row shows the MCDG solutions with the same times T as in the first row, while $C = 0.1$. The third row presents the MCDG solutions with $C = 1.0$. The figure concludes that

when $C = 0.01, 0.1$, which is smaller than the magnitude of each component of $\mathbf{S}$, the MCDG contaminant moves in a clear direction, which is dominated by $\mathbf{S}$. However, when $C = 1.0$, the level of magnitude in the randomness of $\boldsymbol{\lambda}$ is the same as $\mathbf{S}$, the MCDG solution diffuses faster than in the cases $C = 0.01, 0.1$, and the solution does not leave completely the initial location in the direction $(1.0, 1.0)^T$. Instead, part of the solution stays in the same location as the initial one. This phenomena is directly caused by the high level of uncertainty in the convection term, which comes from $\boldsymbol{\lambda}$. Because the convection term determines the moving direction of the solution, the moving direction of the solution shows randomness when it is dominated by $\boldsymbol{\lambda}$. In addition, the diffusion coefficient is far smaller than the convection coefficient. These two factors lead to the difference shown in the third row in Fig 2.5, where the MCDG solutions are dispersive rather than as concentrated as ones in the first and second row.

Approach Two: Deterministic Convection Coefficient b

We know that it is not acceptable to substitute the random convection term with its expected value to do the simulation. However, in this section, I use a numerical example to show that by averaging the velocity is not correct to do the simulation. The numerical example is the same as (2.34), but I take the average of the samples of the convection coefficient $\mathbf{b}$ and put it into this parabolic equation (2.34). We notice that the expected value of the convection coefficients in approach one is equal to $\mathbf{S}$, which is $(1.0, 1.0)^T$. Therefore, I am interested in studying the DG solution to problem (2.34) where the convection coefficient is taken to be the expected value $(1.0, 1.0)^T$. In essence, it becomes a deterministic parabolic equation. By using NIPG with penalty parameter equal to 1.0, I obtain the deterministic DG solutions

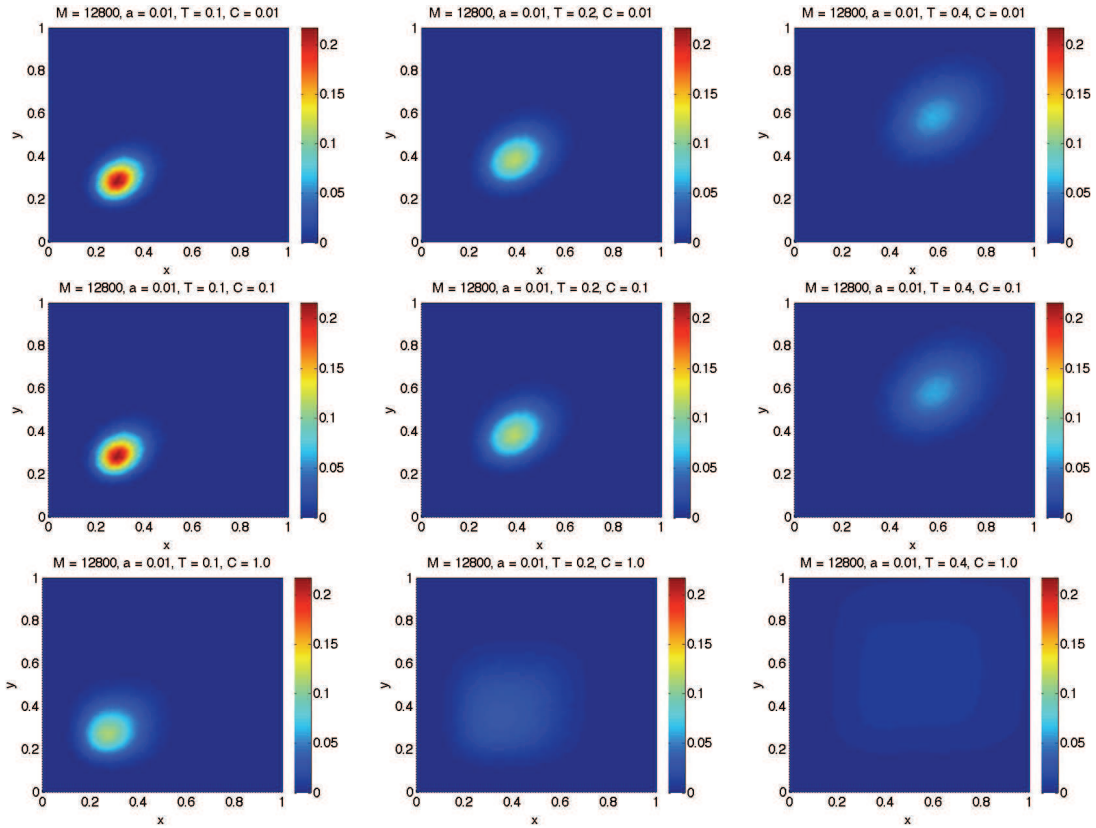


Figure 2.5 : MCDG solutions with $\mathbf{b} = \mathbf{S} + C\boldsymbol{\lambda}, \mathbf{S} = (1.0, 1.0)^T$. Each component of $\boldsymbol{\lambda}$ satisfies standard normal distribution. In the first row, time T is equal to $0.1, 0.2, 0.3$ from left to right and C is equal to 0.01. In the second row, time T is equal to $0.1, 0.2, 0.3$ from left to right and C is equal to 0.1. In the third row, time T is equal to $0.1, 0.2, 0.3$ from left to right and C is equal to 1.0. There are 12800 simulations for each MCDG solution.

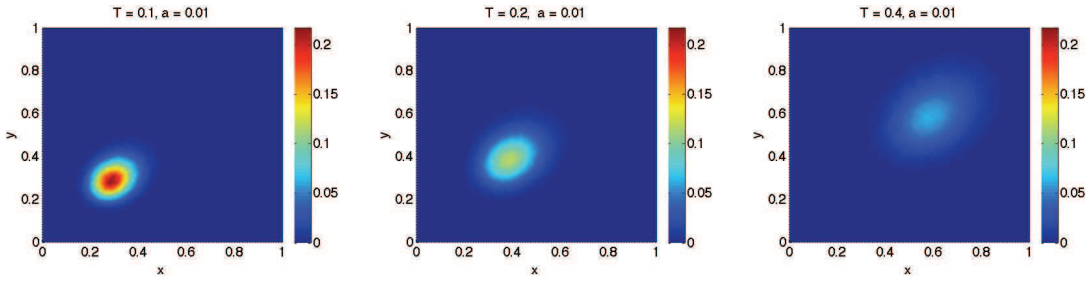


Figure 2.6 : DG solutions with deterministic convection coefficient $\mathbf{b} = (1.0, 1.0)^T$. Time $T = 0.1, 0.2, 0.4$ from left to right.

in Fig 2.6.

By comparing the MCDG solutions in the third row in Fig 2.5 with Fig 2.6, I conclude that when the convection term is randomly distributed, the MCDG solution is more dispersive than DG solution. When the convection term is in a high level of randomness, part of the MCDG solution stays in the same location as the initial one, while the deterministic DG solution leaves completely the initial location in the direction determined by the convection coefficient.

We conclude that it is important to take the high level of uncertainty in the input data into account.

Chapter 3

Generation of Random Permeability

3.1 Introduction

In this chapter, I apply the MCDG method to solve the single phase flow model in a porous medium with random permeability. In section 3.2, I discuss the properties of random permeability of a porous medium. In section 3.3, I present the kernel Principal Component Analysis (PCA) method, which includes linear PCA, kernel PCA and pre-image. I then apply the kernel PCA method to obtain realizations of random permeability fields for two types of porous media. In the next section, the generated permeability realizations are used to approximate the solution to a single phase flow model, on which MCDG is applied.

3.2 Random Permeability

Permeability is the measure of the ability of a porous material to allow fluids to pass through it. Darcy's law states that permeability connects the pressure gradient and the volumetric flow rate of a fluid through the medium. Therefore, knowledge of permeability is of importance in determining the flow characteristics of hydrocarbons in oil and gas reservoir, and of ground water in aquifers. The International System of Units for permeability is m^2 and a traditional unit for permeability is the *darcy* (D), or more commonly the *millidarcy* (mD) (1 darcy $\approx 10^{-12} m^2$). The permeability may vary significantly according to the different types of the porous materials. For

example, the permeability of highly fractured rocks ranges from 10^5 to 10^8 millidarcy, while permeability of granite ranges only from 10^{-3} to 10^{-4} millidarcy. The different permeabilities imply the different conductivities of the porous media. Therefore, an accurate approximation of permeability fields is essential for reservoir simulation in practice. It is also well known that the permeability of porous medium is randomly distributed in earth and shows some correlation structure. In many papers [38–40], permeability is approximated by random fields. Kolyukhin, Espedal [41] and Fenton, Griffiths [42] assumed that the permeability field satisfies a log-normal distribution, i.e., $\log(K)$ is a random variable which satisfies a normal distribution, where K is the permeability field; and they verified this assumption by numerical examples. On the other hand, the correlation structure in permeability fields is usually described by *correlation length* [43–45], which is a distance from a point beyond which there is no correlation of a physical property associated with that point. If the distance between two points is larger than correlation points, they are not correlated. Taking the layering effect into account, Hinkelmann [45] pointed out that the horizontal correlation is 5 or 10 times larger than the vertical one. Zhang and Lu [46] used Karhunem–Loève expansion to approximate the permeability fields and showed that if the correlation length is large, only a few terms in K–L expansion are required to approximate the permeability field; if the correlation length is relatively small, a large number of terms is needed to reach a certain accuracy. In the following section, I study the kernel PCA method [31], which is able to reproduce the complex geological structure, such as channels, very well numerically and requires the eigen-decomposition of the kernel matrix generated by a very small set of realizations in geological simulations.

3.3 Kernel Principal Component Analysis (PCA)

It is well known that Karhunen–Loève expansion can be used as a differential representation of random fields which appear in geological models. However, the standard K–L expansion suffers two major drawbacks: first, it requires an eigen-decomposition of the covariance matrix of the random field, which is very expensive for a large scale problem associated with a large covariance matrix; second, it preserves only the two-point statistics of a random field, which is not sufficient to generate random fields with complex structures, such as channels. Kernel Principal Component Analysis (PCA) is applied to address these two limitations of the standard K–L expansion, because it only requires the eigen-decomposition of the kernel matrix generated by a set of realizations in geological simulations and perform very well when reproduction of complex geological structures, such as channels, are required. To elaborate, let N_h denote the number of cells in the geological model, and let N_r denote the number of realizations required to obtain a converged covariance matrix. A covariance matrix is said to be convergent if the covariance between elements at any cell i and j is convergent as the number of simulations increases. The size of the covariance matrix in K–L expansion is $N_h \times N_h$, while the size of the kernel matrix in kernel PCA is $N_r \times N_r$. In the paper [47], the authors claimed that in general, a value of N_r of order 10^2 to 10^3 should be sufficient for problems with N_h of the order 10^5 to 10^6.

In this thesis, I use polynomial kernel PCA of order two to generate random permeability fields. The data used to build the kernel matrix comes from the SPE10 model, which is described in the following section.

3.3.1 SPE10 Model

The SPE10 model is the Tenth SPE Comparative Solution Project [33]. It was designed to compare the performances of different simulators and algorithms by providing the benchmark datasets. Two datasets are created in this model. The first one is a two dimensional vertical section, which yields a small set of data. The second one is a three dimensional model with millions of hexaedra cells. The values of permeability and porosity are provided for each of these cells. In my thesis, I use the permeability values in the second dataset in the SPE10 model to generate random permeability fields by kernel PCA. To clarify, when I mention SPE10 model in my thesis, I mean the second dataset in the SPE10 model.

The model of second dataset is described on a regular Cartesian grid. The dimensions of this model are $1200 \times 2200 \times 170$(ft). The fine scale cell size is $20 \times 10 \times 2$ (ft). Thus, there are 85 layers in this model. The top 35 layers represent the porous medium of Tarbert formation, where there is no channel structure. The bottom 50 layers represent the porous medium of Upper Ness formation, where channel structures appear. The physical domain of each layer is $[0, 1200] \times [0, 2200] \times [0, 2]$. For simplicity, I consider the permeability data for the region $[0, 1200] \times [0, 600]$ of each layer, and use this restricted data to generate random permeability fields by kernel PCA. Since the size of each cell is 20×10 (ft) for each layer, the region is partitioned into 60×60 cells. After transforming the rectangular cells into square cells with length of each side equal to 20 (ft) and converting the unit from feet to meter, I obtain that the physical domain in the following numerical experiments is $D = (0, 360) \times (0, 360)$ (meter). To clarify, when I mention the grids of size $60 \times 60, 30 \times 30, 15 \times 15$ in the following, I mean that the domain contains $60 \times 60, 30 \times 30, 15 \times 15$ cells and the size of the domain is always $(0, 360) \times (0, 360)$ (meter).

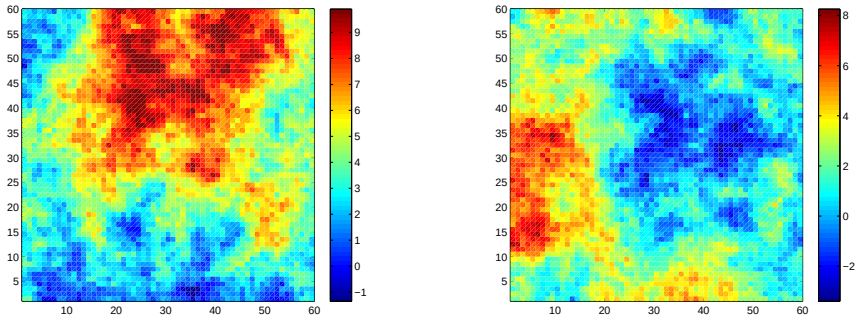


Figure 3.1 : Natural logarithm of permeability field of layer 4 (left) and layer 31 (right) in the SPE10 model.

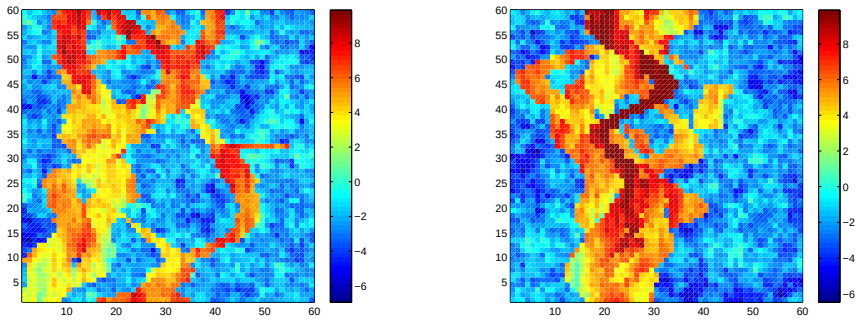


Figure 3.2 : Natural logarithm of permeability field of layer 49 (left) and layer 79 (right) in the SPE10 model.

The permeabilities in the top 35 layers are used to generate realizations of a random permeability field for the porous medium of Tarbert formation. Similarly, the permeabilities in the bottom 50 layers are used to generate realizations of a random permeability field for a porous medium of Upper Ness formation. We show in Fig 3.1 the natural logarithm of permeabilities in layer 4 and layer 31 in the SPE10 model. Fig 3.2 presents the natural logarithm of permeabilities in layer 49 and layer 79 in the SPE10 model.

3.3.2 Linear PCA

In this section, I discuss the linear PCA method and apply it to derive a differentiable representation of a random permeability field $\mathbf{S}$. Essentially, the formula of the random field generated by linear PCA is the same as the one generated by K-L expansion. We start with the covariance matrix generated by a set of discrete centered (i.e. with zero mean) realizations of a random field $\bar{\mathbf{y}}_j, j = 1, \cdots, N_r, \mathbf{y}_j \in \mathbb{R}^{N_h}$, where

$$\bar{\mathbf{y}}_j = \mathbf{y}_j - \frac{1}{N_r} \sum_{k=1}^{N_r} \mathbf{y}_k. \tag{3.1}$$

Here, N_r is the number of realizations $\mathbf{y}_j$, which are given to generate kernel matrix, and N_h is the number of cells in the domain. The covariance matrix can be calculated as:

$$\mathbf{C} = \frac{1}{N_r} \sum_{j=1}^{N_r} \bar{\mathbf{y}}_j \bar{\mathbf{y}}_j^T. \tag{3.2}$$

The size of $\mathbf{C}$ is $N_h \times N_h$. Let $(\lambda, \mathbf{v})$ be an eigenpair of $\mathbf{C}$, then

$$\lambda \mathbf{v} = \mathbf{C}\mathbf{v} = \frac{1}{N_r} \sum_{j=1}^{N_r} \bar{\mathbf{y}}_j (\bar{\mathbf{y}}_j^T \mathbf{v}). \tag{3.3}$$

This implies that the eigenvectors $\mathbf{v}$ associated with nonzero eigenvalues belong to $\text{span}\{\bar{\mathbf{y}}_1, \cdots, \bar{\mathbf{y}}_{N_r}\}$. This indicates that there are at most N_r distinct non-zero eigenvalues associated with N_r eigenvectors $\mathbf{v}$. Let $\mathbf{v}_1, \cdots, \mathbf{v}_{N_r}$ be the eigenvectors. Expand

$$\mathbf{v}_i = \sum_{j=1}^{N_r} \alpha_j^i \bar{\mathbf{y}}_j \quad i = 1, \cdots, N_r. \tag{3.4}$$

Let $\bar{\mathbf{Y}} = [\bar{\mathbf{y}}_1 \mid \cdots \mid \bar{\mathbf{y}}_{N_r}]$ and define $\boldsymbol{\alpha}^i = [\alpha_1^i, \cdots, \alpha_{N_r}^i]^T$. From equation (3.4), we have

$$\mathbf{v}_i = \bar{\mathbf{Y}} \boldsymbol{\alpha}^i \quad i = 1, \cdots, N_r.$$

so that

$$\mathbf{v}_i^T \mathbf{v}_j = \boldsymbol{\alpha}^{iT} \bar{\mathbf{Y}}^T \bar{\mathbf{Y}} \boldsymbol{\alpha}^j. \tag{3.5}$$

In the following, I derive the kernel eigenvalue problem and show the orthogonality of $\{\mathbf{v}_i\}$ for certain $\{\boldsymbol{\alpha}_i\}$.

From equation (3.3), we have

$$\lambda(\bar{\mathbf{y}}_i^T \mathbf{v}_i) = \bar{\mathbf{y}}_i^T (\mathbf{C}\mathbf{v}_i) \quad \forall i = 1, \cdots, N_r. \tag{3.6}$$

Plug (3.4) into (3.6), we have

$$\lambda \sum_{j=1}^{N_r} \alpha_j^i \left(\bar{\mathbf{y}}_i^T \bar{\mathbf{y}}_j\right) = \frac{1}{N_r} \sum_{j=1}^{N_r} \alpha_j^i \bar{\mathbf{y}}_i^T \left(\sum_{k=1}^{N_r} \left(\bar{\mathbf{y}}_i^T \bar{\mathbf{y}}_k\right) \left(\bar{\mathbf{y}}_k^T \bar{\mathbf{y}}_j\right) \right) \quad \forall i = 1, \cdots, N_r. \tag{3.7}$$

Define an $N_r \times N_r$ matrix $\mathbf{K}$ where $K_{ij} = \bar{\mathbf{y}}_i^T \bar{\mathbf{y}}_j, i, j = 1, \cdots, N_r$. The matrix $\mathbf{K}$ is called *kernel matrix*. Equation (3.7) can be written as:

$$N_r \lambda \mathbf{K} \boldsymbol{\alpha} = \mathbf{K}^2 \boldsymbol{\alpha}. \tag{3.8}$$

Compared with the size of covariance matrix $\mathbf{C}$ which is $N_h \times N_h$, the size of the kernel matrix is $N_r \times N_r$. Here, each entry $K_{ij} = \bar{\mathbf{y}}_i^T \bar{\mathbf{y}}_j$ is called a *polynomial kernel of order one*. Since $\mathbf{K}$ is a symmetric positive definite matrix and the eigenvectors of $\mathbf{K}$ are the same as the ones of $\mathbf{K}^2$, and the eigenvalues of $\mathbf{K}^2$ are the squares of the eigenvalues of $\mathbf{K}$, thus, it is sufficient to find the solution to the eigen-problem (3.8) by solving:

$$N_r \lambda \boldsymbol{\alpha} = \mathbf{K} \boldsymbol{\alpha}. \tag{3.9}$$

Problem (3.9) is known as the kernel eigenvalue problem with the eigenpair $(N_r\lambda, \boldsymbol{\alpha})$. By solving (3.9), we can obtain the eigenvalue of problem (3.3) by dividing N_r on the eigenvalue of problem (3.9). The eigenvector $\mathbf{v}$ in problem (3.3) can be obtained from $\boldsymbol{\alpha}$ by using (3.4).

The advantage of solving the kernel eigenvalue problem (3.9) is that the dimension of kernel matrix $\mathbf{K}$ is only $N_r \times N_r$ which is significantly less than $N_h \times N_h$. Therefore, it is more efficient.

Since $\mathbf{K} = \bar{\mathbf{Y}}^T\bar{\mathbf{Y}}$, we have

$$\bar{\mathbf{Y}}^T\bar{\mathbf{Y}}\boldsymbol{\alpha}^i = \mathbf{K}\boldsymbol{\alpha}^i = N_r\lambda_i\boldsymbol{\alpha}^i \quad i = 1, \cdots, N_r. \tag{3.10}$$

Plug (3.10) into (3.5), we have

$$\mathbf{v}_i^T\mathbf{v}_j = N_r\lambda_j{\boldsymbol{\alpha}^i}^T\boldsymbol{\alpha}^j = \begin{cases} 0 & \text{if} \quad i \neq j \\ N_r\lambda_j & \text{if} \quad i = j, \end{cases} \tag{3.11}$$

for orthonormal $\{\boldsymbol{\alpha}_i\}, i = 1, \cdots, N_r$. Since $\{\boldsymbol{\alpha}_i\}$ are the eigenvectors of kernel matrix $\mathbf{K}$, which is symmetric positive semidefinite, we can find such orthonormal eigenvectors $\{\boldsymbol{\alpha}_i\}, i = 1, \cdots, N_r$.

This implies that the $\mathbf{v}_i$'s are orthogonal. By normalizing them, we obtain

$$\mathbf{v}_i = \frac{1}{\sqrt{N_r\lambda_i}}\sum_{j=1}^{N_r}\alpha_j^i\bar{\mathbf{y}}_j \quad i = 1, \cdots, N_r. \tag{3.12}$$

The truncated K–L expansion, Theorem 2.2, suggests the following expansion for each component of the vector of the random permeability field $\mathbf{S}(w, x)$

$$\mathbf{S}(w, x_i) = \frac{1}{N_r}\sum_{j=1}^{N_r}\mathbf{y}_{ji} + \sum_{j=1}^{N_r}\sqrt{\lambda_j}\phi_j(w)\mathbf{v}_{ji} \tag{3.13}$$

where $\mathbf{v}_{ji}$ and $\mathbf{y}_{ji}$ are the i-th component of vector $\mathbf{v}_j$ and $\mathbf{y}_j$ respectively, ϕ_j are uncorrelated random variables with mean value zero and unit variance. In this thesis, I use uniform distribution in the interval $[-\sqrt{3}, \sqrt{3}]$ to sample ϕ_j. The reason to take the interval $[-\sqrt{3}, \sqrt{3}]$ is that the mean of ϕ_j is zero and the variance of ϕ_j is one as ϕ_j satisfies uniform distribution in $[-\sqrt{3}, \sqrt{3}]$.

The algorithm of linear PCA to generate random permeability field is:

1. Realizations of random permeability $\mathbf{y}_1, \cdots, \mathbf{y}_{N_r}$ are given.

2. Obtain the centered realizations

$$\bar{\mathbf{y}}_j = \mathbf{y}_j - \frac{1}{N_r}\sum_{j=1}^{N_r}\mathbf{y}_j. \tag{3.14}$$

3. Compute the kernel matrix $\mathbf{K}$, where $K_{ij} = \bar{\mathbf{y}}_i^T\bar{\mathbf{y}}_j$.

4. Perform the eigen-decomposition of $\mathbf{K}$. Obtain eigenvectors $\{\boldsymbol{\alpha}^i\}$ associated with nonzero eigenvalues $\{\mu_i\}$ of $\mathbf{K}$, where $\mu_i = N_r\lambda_i$ and λ_i is the eigenvalue of the covariance matrix $\mathbf{C}$.

5. Compute eigenvectors of covariance matrix $\mathbf{C}$

$$\mathbf{v}_i = \bar{\mathbf{Y}}\boldsymbol{\alpha}^i \quad i = 1, \cdots, N_r. \tag{3.15}$$

   where $\bar{\mathbf{Y}} = [\bar{\mathbf{y}}_1 \mid \cdots \mid \bar{\mathbf{y}}_{N_r}]$

6. Sample one realization of random variables ϕ_j which satisfies uniform distribution in the interval $[-\sqrt{3}, \sqrt{3}]$, so mean of ϕ_j is zero and variance of ϕ_j is one.

7. Compute one realization of random permeability

$$\mathbf{S} = \frac{1}{N_r}\sum_{j=1}^{N_r}\mathbf{y}_j + \sum_{j=1}^{N_r}\sqrt{\frac{\mu_j}{N_r}}\phi_j\mathbf{v}_j \tag{3.16}$$

8. By repeating step 6 and 7 with another realization of ϕ_j, we obtain another realization of random permeability $\mathbf{S}$.

Essentially, linear PCA is the discrete version of K-L expansion. In Table 3.1, I compare the linear PCA and K-L expansion and point out their relationship.

Table 3.1 : Comparison of linear PCA and K-L expansion

	K-L expansion	Linear PCA
Matrix	Covariance: $\mathbf{C} = \frac{1}{N_r}\sum_{j=1}^{N_r} \bar{\mathbf{y}}_j \bar{\mathbf{y}}_j^T$	Kernel, each entry $K_{ij} = \bar{\mathbf{y}}_i^T \bar{\mathbf{y}}_j$
Eigen-decomp	$\lambda \mathbf{v} = \mathbf{C}\mathbf{v}$	$\mu \boldsymbol{\alpha} = \mathbf{K}\boldsymbol{\alpha} = N_r \lambda \boldsymbol{\alpha}$
Relationship	$\mathbf{v} = \bar{\mathbf{Y}}\boldsymbol{\alpha}, \quad \lambda = \frac{\mu}{N_r}, \quad$ where $\bar{\mathbf{Y}} = [\bar{\mathbf{y}}_1 \mid \cdots \mid \bar{\mathbf{y}}_{N_r}]$	
Random Field	$\mathbf{S} = \frac{1}{N_r}\sum_{j=1}^{N_r} \mathbf{y}_j + \sum_{j=1}^{N_r} \sqrt{\lambda_j}\phi_j \mathbf{v}_j = \frac{1}{N_r}\sum_{j=1}^{N_r} \mathbf{y}_j + \sum_{j=1}^{N_r} \sqrt{\frac{u_j}{N_r}}\phi_j \left(\bar{\mathbf{Y}}\boldsymbol{\alpha}^j\right)$	

In the end of this section, I clarify that the word "preserve" means that the mean and covariance of the generated random field are the same as the original one. If we use PCA, we must have a covariance matrix or samples from the random field, so that we can use them to create a covariance matrix $\mathbf{C}$, like (3.2). Since $\mathbf{C}$ is positive semi-definite, we have eigen-decomposition

$$\mathbf{C} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{V}^T \tag{3.17}$$

where $\mathbf{V}\mathbf{V}^T = \mathbf{I}$. By Theorem 2.2, we know that the random field $\mathbf{S}$ can be generated by truncated K-L expansion and is written as

$$\mathbf{S} = \mathbf{V}\boldsymbol{\Sigma}^{1/2}\boldsymbol{\phi} \tag{3.18}$$

where $\boldsymbol{\phi} = (\phi_1, \cdots, \phi_{N_h})$ and $\phi_i, i = 1, \cdots, \phi_{N_h}$ satisfies uniform distribution with mean 0 and unit variance. Formula (3.18) also appears in Sarma, Durlofsky,and Aziz's paper [47] as formula (2). Thus, the covariance matrix of the random vector $\boldsymbol{\phi}$ is an identity matrix $\mathbf{I} \in \mathbb{R}^{N_h \times N_h}$. Since the matrices $\mathbf{V}$ and $\boldsymbol{\Sigma}$ are constant, the covariance matrix of the generated random field $\mathbf{S}$ is

$$\mathbf{C}(\mathbf{S}) = \mathbf{V}\boldsymbol{\Sigma}^{1/2}\boldsymbol{\Sigma}^{T\,1/2}\mathbf{V}^T\mathbf{I} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{V}^T, \tag{3.19}$$

which is the same as the covariance matrix of the original random field. Since the mean of $\boldsymbol{\phi}$ is equal to zero, the mean of $\mathbf{S}$ is also equal to zero by (3.18). Therefore, linear PCA preserves the same mean and covariance as of the original random field. However, we cannot show that linear PCA preserves the higher-order moments of the original random field. Since a Gaussian model (two-point statistics) is defined by the covariance matrix, it follows that linear PCA works for Gaussian models. However, when the random field is characterized by multipoint geostatistics (e.g., channelized models), higher-order moments are needed to define them. The direct use of linear PCA leads to Gaussian looking models, rather than the multipoint geostatistic models which we expect to capture. In section 4 of the paper [47], Sarma, Durlofsky and Aziz show this conclusion.

In the following section, I will discuss Kernel PCA, which is a method which is able to capture the geological structure characterized by multipoint geostatistics.

3.3.3 Kernel PCA

We recall above that standard K–L expansion only preserves the two-point statistics of the random fields, which is not able to reproduce complex geological structures, such as channels. In this section, we present the higher order kernel PCA which is a method of multipoint statistics and is shown to be able to capture channel structures [47] numerically.

Consider a non-linear mapping Ψ, for $L \gg N_h$:

$$\Psi : \mathbb{R}^{N_h} \to \mathbb{R}^L. \tag{3.20}$$

In the literature [31, 47], the image of Ψ is denoted by F, and is called the feature space. By this transformation, the realizations which are non-linearly related in $\mathbb{R}^{N_h}$

become linearly related in $\mathbb{R}^L$. Then, the standard K–L expansion can be applied in $\mathbb{R}^L$. The results in the previous section can be generalized for general kernel PCA. Let $\mathbf{y}_j \in \mathbb{R}^{N_h}, j = 1, \cdots, N_r$. We center $\Psi(\mathbf{y}_j)$ and denote it by $\bar{\Psi}(\mathbf{y}_j)$:

$$\bar{\Psi}(\mathbf{y}_j) = \Psi(\mathbf{y}_j) - \frac{1}{N_r} \sum_{k=1}^{N_r} \Psi(\mathbf{y}_k).$$

The covariance matrix is:

$$\mathbf{C} = \frac{1}{N_r} \sum_{j=1}^{N_r} \bar{\Psi}(\mathbf{y}_j) \bar{\Psi}(\mathbf{y}_j)^T, \tag{3.21}$$

where the dimension of this covariance matrix is not $N_h \times N_h$, but $L \times L$, where L could be extremely large. The eigenvalue problem in $\mathbb{R}^L$ is:

$$\lambda \mathbf{v} = \mathbf{C}\mathbf{v}, \tag{3.22}$$

where $(\lambda, \mathbf{v})$ is an eigenpair of $\mathbf{C}$. As in the previous section, we do not directly solve this eigenvalue problem because of the extremely large dimension of the covariance matrix $\mathbf{C}$. Instead, we solve the kernel eigenvalue problem:

$$N_r \lambda \boldsymbol{\alpha} = \mathbf{K}\boldsymbol{\alpha}. \tag{3.23}$$

where $K_{ij} = \bar{\Psi}(\mathbf{y}_i)^T \bar{\Psi}(\mathbf{y}_j)$. We can obtain the eigenpair $(\lambda, \mathbf{v})$ of covariance matrix $\mathbf{C}$ by the eigendecomposition of problem (3.23). It is similar to the procedure in the previous section. We define a kernel function k as below:

$$k(\mathbf{y}_i, \mathbf{y}_j) = \bar{\Psi}(\mathbf{y}_i)^T \bar{\Psi}(\mathbf{y}_j). \tag{3.24}$$

The kernel function $k(\mathbf{x}, \mathbf{y})$ calculates the dot product in space $\mathbb{R}^L$ directly from the element $\mathbf{y}_i \in \mathbb{R}^{N_h}$. In my thesis, k is chosen to be:

$$k(\mathbf{y}_i, \mathbf{y}_j) = \mathbf{y}_i^T \mathbf{y}_j + (\mathbf{y}_i^T \mathbf{y}_j)^2. \tag{3.25}$$

This means that Ψ takes the form:

$$\Psi(\mathbf{x}) = \left(x_1, \cdots, x_{N_h}, x_1^2, \cdots, x_{N_h}^2, \cdots, \sqrt{2}x_i x_j, \cdots\right), \tag{3.26}$$

where we choose $1 \leq i < j \leq N_h$ for the terms of $\sqrt{2}x_i x_j$. In this case, the dimension of the feature space $\mathbb{R}^L$ is

$$L = \frac{(N_h + 1)N_h}{2}. \tag{3.27}$$

Pre-image

For the kernel PCA, K–L expansion is performed in the feature space $\mathbb{R}^L$. The resulting $\hat{\mathbf{y}}$ belongs to $\mathbb{R}^L$. However, we are interested in obtaining the simulation result $\mathbf{y}$ in the original space $\mathbb{R}^{N_h}$. That means we need to solve the pre-image problem [31, 48].

$$\mathbf{y} = \Psi^{-1}(\hat{\mathbf{y}}). \tag{3.28}$$

Since the dimension L can be extremely large, it may not be possible to compute the pre-image or the pre-image may not exist. Even if it exists, the authors in [31, 48] point out that the solution may not be unique. Due to these reasons, we address this problem by solving a least-square problem as below:

$$\min_{\mathbf{y} \in \mathbb{R}^{N_h}} ||\Psi(\mathbf{y}) - \hat{\mathbf{y}}||^2 \quad \text{where} \quad ||\Psi(\mathbf{y}) - \hat{\mathbf{y}}||^2 = \Psi(\mathbf{y})^T\Psi(\mathbf{y}) - 2\hat{\mathbf{y}}^T\Psi(\mathbf{y}) + \hat{\mathbf{y}}^T\hat{\mathbf{y}}. \tag{3.29}$$

Now, we derive a representation of the random variable $\hat{\mathbf{y}}$ in the feature space. Recall the definition of the covariance matrix $\mathbf{C}$:

$$\mathbf{C} = \frac{1}{N_r}\sum_{j=1}^{N_r} \bar{\Psi}(\mathbf{y}_j)\bar{\Psi}(\mathbf{y}_j)^T. \tag{3.30}$$

The eigenvalue problem in $\mathbb{R}^L$ is:

$$\lambda\mathbf{v} = \mathbf{C}\mathbf{v}. \tag{3.31}$$

By (3.30) and (3.31), we have

$$\mathbf{v} = \frac{1}{\lambda N_r} \sum_{j=1}^{N_r} \left(\bar{\Psi}(\mathbf{y}_j)^T \mathbf{v} \right) \bar{\Psi}(\mathbf{y}_j). \tag{3.32}$$

Therefore, for each eigenvector $\mathbf{v}_i, i = 1, \cdots, N_r$, we obtain coefficients $(\alpha_j^i)_j$ such that

$$\mathbf{v}_i = \sum_{j=1}^{N_r} \alpha_j^i \bar{\Psi}(\mathbf{y}_j). \tag{3.33}$$

This can be written as

$$\mathbf{v}_i = \bar{\Psi}(\mathbf{Y}) \boldsymbol{\alpha}^i, \tag{3.34}$$

where $\bar{\Psi}(\mathbf{Y}) = [\bar{\Psi}(\mathbf{y}_1) \mid \cdots \mid \bar{\Psi}(\mathbf{y}_{N_r})]$ is a matrix of size $L \times N_r$, and $\boldsymbol{\alpha}^i = (\alpha_1^i, \cdots, \alpha_{N_r}^i)$. The truncated K-L expansion (see Theorem 2.2) suggests

$$\hat{\mathbf{y}} = \mathbf{V} \Lambda^{\frac{1}{2}} \boldsymbol{\phi},$$

where the columns of the matrix $\mathbf{V} = [\mathbf{v}_1 \mid \cdots \mid \mathbf{v}_{N_r}]$ are the eigenvectors of the covariance matrix $\mathbf{C}$, Λ is a diagonal matrix filled with the eigenvalues $\{\lambda_i\}$ of $\mathbf{C}$, and $\boldsymbol{\phi} = (\phi_1, \cdots, \phi_{N_r})$ is a vector of uncorrelated random variables. Thus, from (3.34) we have

$$\begin{aligned}
\hat{\mathbf{y}} &= [\mathbf{v}_1 \mid \cdots \mid \mathbf{v}_{N_r}] \Lambda^{\frac{1}{2}} \boldsymbol{\phi} \\
&= [\bar{\Psi}(\mathbf{Y}) \boldsymbol{\alpha}^1 \mid \cdots \mid \bar{\Psi}(\mathbf{Y}) \boldsymbol{\alpha}^{N_r}] \Lambda^{\frac{1}{2}} \boldsymbol{\phi} \\
&= \bar{\Psi}(\mathbf{Y}) [\boldsymbol{\alpha}^1 \mid \cdots \mid \boldsymbol{\alpha}^{N_r}] \Lambda^{\frac{1}{2}} \boldsymbol{\phi} \\
&= \sum_{j=1}^{N_r} \beta_j \bar{\Psi}(\mathbf{y}_j).
\end{aligned}$$

Since the eigenvalue μ_i of the kernel matrix $\mathbf{K}$ is equal to $\lambda_i N_r$, the coefficient β_i can be computed as:

$$\beta_i = \frac{1}{\sqrt{N_r}} \sum_{j=1}^{N_r} \alpha_j^i \sqrt{\mu_j} \phi_j \quad i = 1, \cdots, N_r. \tag{3.35}$$

Therefore, we can approximate $\hat{\mathbf{y}}$ in the feature by

$$\hat{\mathbf{y}} = \frac{1}{\sqrt{N_r}} \sum_{i=1}^{N_r} \sum_{j=1}^{N_r} \alpha_j^i \sqrt{\mu_j} \phi_j \bar{\Psi}(\mathbf{y}_i), \tag{3.36}$$

where $\boldsymbol{\alpha}^i = (\alpha_1^i, \cdots, \alpha_{N_r}^i)^T$ and μ_i are the ith eigenvector and eigenvalue of the kernel matrix $\mathbf{K}$, respectively, and $\{\phi_j\}$ are uncorrelated random variables.

By using (3.36) and replacing the dot product with the kernel function defined in (3.24), we can write problem (3.29) as:

$$||\Psi(\mathbf{y}) - \hat{\mathbf{y}}||^2 = k(\mathbf{y}, \mathbf{y}) - 2 \sum_{i=1}^{N_r} \beta_i k(\mathbf{y}_i, \mathbf{y}) + \sum_{i=1}^{N_r} \sum_{j=1}^{N_r} \beta_i \beta_j k(\mathbf{y}_i, \mathbf{y}_j). \tag{3.37}$$

Using the formula of kernel function $k(\mathbf{y}, \mathbf{y})$ in (3.25) and referring to the scheme in section 5 in the paper [47] to solve the minimization problem (3.37), we have the following iteration scheme:

$$\mathbf{y}^{k+1} = \frac{\sum_{i=1}^{N_r} \beta_i \sum_{j=1}^{2} j \cdot (\mathbf{y}_i^T \mathbf{y}^k)^{j-1} \mathbf{y}_i}{\sum_{j=1}^{2} j \cdot (\mathbf{y}^{kT} \mathbf{y}^k)^{j-1}}. \tag{3.38}$$

However, this iteration scheme is not stable for polynomial kernel functions. In paper [47], the iteration in (3.38) is modified to

$$\mathbf{y}^{k+1} = \frac{\sum_{i=1}^{N_r} \beta_i \sum_{j=1}^{2} j \cdot (\mathbf{y}_i^T \mathbf{y}^k)^{j-1} \mathbf{y}_i}{\sum_{i=1}^{N_r} \beta_i \sum_{j=1}^{2} j \cdot (\mathbf{y}_i^T \mathbf{y}^k)^{j-1}}. \tag{3.39}$$

which is our final iteration scheme.

To summarize, the kernel PCA algorithm to generate random permeability fields is:

1. Realizations of random permeability $\mathbf{y}_1, \cdots, \mathbf{y}_{N_r}$ are given.

2. Compute the image $\Psi(\mathbf{y}_j), j = 1, \cdots, N_r$, with Ψ defined by (3.26).

3. Center $\Psi(\mathbf{y}_j), j = 1, \cdots, N_r$:

$$\bar{\Psi}(\mathbf{y}_j) = \Psi(\mathbf{y}_j) - \frac{1}{N_r} \sum_{k=1}^{N_r} \Psi(\mathbf{y}_k). \tag{3.40}$$

4. Compute the kernel matrix $\mathbf{K}$, where $K_{ij} = \bar{\Psi}(\mathbf{y}_i)^T \bar{\Psi}(\mathbf{y}_j)$

5. Perform the eigen-decomposition of $\mathbf{K}$. Obtain eigenvectors $\{\boldsymbol{\alpha}^i\}$ associated with nonzero eigenvalues $\{\mu_i\}$ of $\mathbf{K}$, where $\mu_i = N_r \lambda_i$ and λ_i is the eigenvalue of the covariance matrix $\mathbf{C}$.

6. Sample one realization of random variables ϕ_j which satisfies uniform distribution in the interval $[-\sqrt{3}, \sqrt{3}]$, so mean of ϕ_j is zero and variance of ϕ_j is one.

7. Compute
$$\beta_i = \frac{1}{\sqrt{N_r}} \sum_{j=1}^{N_r} \alpha_j^i \sqrt{\mu_j} \phi_j \tag{3.41}$$

8. Initialize $\mathbf{y}^1 = \mathbf{y}_1$, where $\mathbf{y}_1$ is one given realization in the first step.

9. Set the maximum number of iterations to be n and the tolerance tol

10. Perform the fixed-point iteration

    - While $||\mathbf{y}^{k+1} - \mathbf{y}^k|| \leq tol$ or $k \leq n-1$ Do
      - formula (3.39)
    - End

11. Obtain one realization of random permeability $\mathbf{S} = \mathbf{y}^n$.

3.3.4 Numerical Examples

Spectrum of Kernel Matrix in Linear PCA and kernel PCA

In this section, I plot the spectrum of the kernel matrix in linear PCA and kernel PCA methods. For each method, I present the spectrum of the kernel matrices for

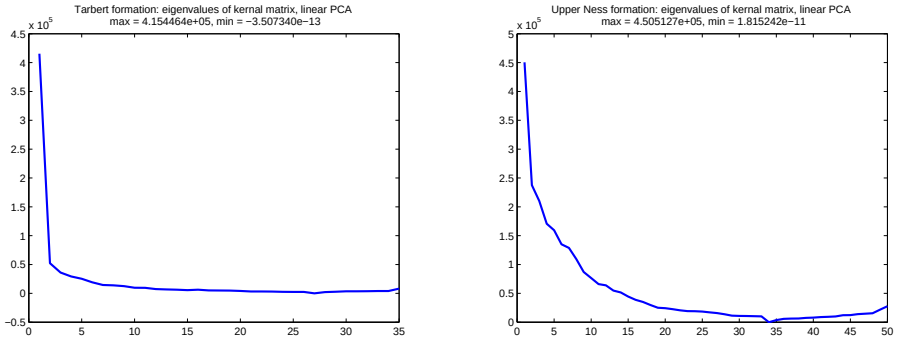


Figure 3.3 : Spectrum of kernel matrices by linear PCA: the left plot is the spectrum of the kernel matrix for the porous medium of Tarbert formation, and the right plot is the spectrum of the kernel matrix for the porous medium of Upper Ness formation.

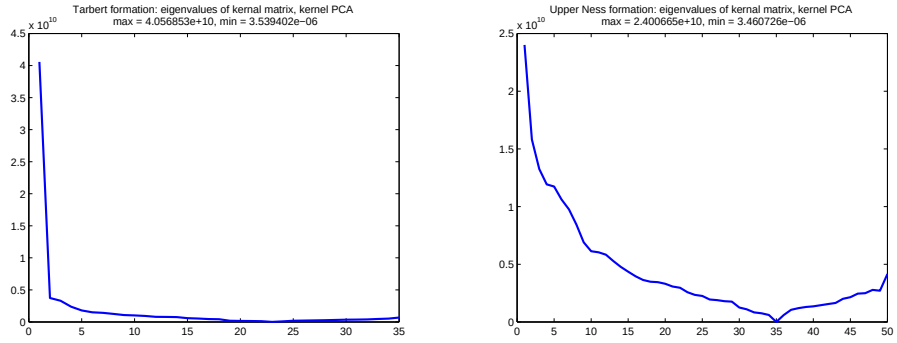


Figure 3.4 : Spectrum of kernel matrices by polynomial kernel PCA of second order: the left plot is the spectrum of the kernel matrix for the porous medium of Tarbert formation, and the right plot is the spectrum of the kernel matrix for the porous medium of Upper Ness formation.

the porous media of Tarber formation and Upper Ness formation. The permeabilities in SPE10 model, which is described in section 3.3.1, are used to created the kernel matrices for both types of porous media.

Generation of Random Permeability by Linear PCA

The permeability values from the SPE10 data are piecewise constants. Denote by $D_1, \cdots, D_{N_h}$ the rectangular cells, and $\bigcup_{p=1}^{N_h} D_p = D, D_p \cap D_q = \emptyset$ if $p \neq q$ for

$q, p = 1, \cdots, N_h$. Define $\tau_p(x)$ by:

$$\forall p = 1, \cdots, N_r \quad \tau_p(x) = \begin{cases} 1 & \text{if} \quad x \in D_p; \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, the random vector field can be written as:

$$\forall 1 \leq i \leq N_r \quad \bar{\mathbf{y}}_i(x) = \sum_{p=1}^{N_h} \gamma_p^i \boldsymbol{\tau}_p(x), \tag{3.42}$$

where γ_p^i is the permeability value at the p^{th} cell, which has been centered:

$$\forall 1 \leq p \leq N_h, \quad \forall 1 \leq i \leq N_r, \quad \gamma_p^i = \gamma_p^i - \frac{1}{N_r} \sum_{j=1}^{N_r} \gamma_p^i. \tag{3.43}$$

Then the kernel matrix is:

$$K_{ij} = \bar{\mathbf{y}}_i^T \bar{\mathbf{y}}_j = \sum_{q=1}^{N_h} \sum_{p=1}^{N_h} \gamma_q^i \gamma_p^j \tau_q^T \tau_p = \left(\boldsymbol{\gamma}^i\right)^T \mathbf{B} \boldsymbol{\gamma}^j \tag{3.44}$$

where $\mathbf{B}$ is a matrix of size $N_h \times N_h$, defined by $B_{qp} = \tau_q^T \tau_p$ for each entry. By definition of τ_p, we have

$$\mathbf{B} = \mathbf{I}.$$

The kernel matrix simply becomes:

$$K_{ij} = \left(\boldsymbol{\gamma}^i\right)^T \boldsymbol{\gamma}^j. \tag{3.45}$$

Then, we apply steps 4-8 of algorithm of linear PCA to generate one realization of a random permeability field for a porous medium of Tarbert formation. The permeability data of the top 35 layers in the SPE10 model is used to generate $\bar{\mathbf{y}}_i$, i.e., $N_r = 35$. Fig 3.5 shows four realizations of the random permeability field generated by linear PCA.

Second, we take $N_r = 50$ and use the permeabilities of bottom 50 layers in the SPE10 model to generate the random permeability field for a porous medium of Upper

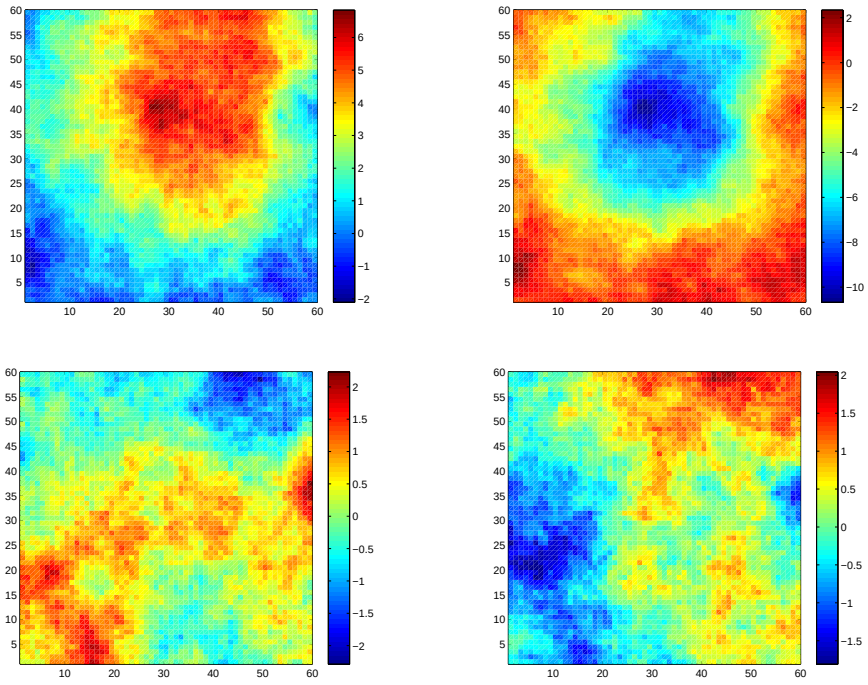


Figure 3.5 : Four realizations of natural logarithm of permeability fields generated by linear PCA method for the porous medium of Tarbert formation. Permeabilities of top 35 layers are used to create the kernel matrix.

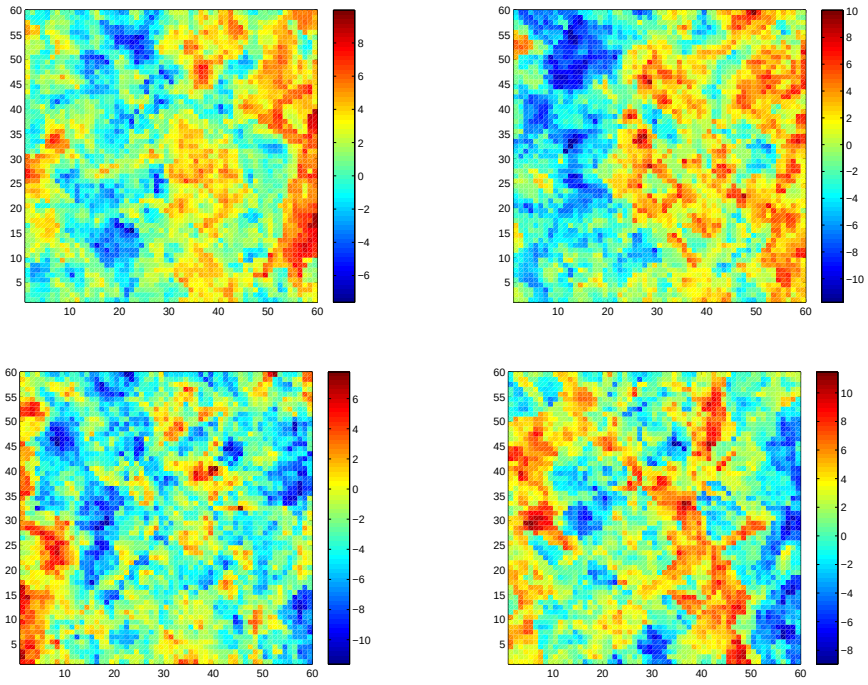


Figure 3.6 : Four realizations of natural logarithm of permeability fields generated by linear PCA method for the porous medium of Upper Ness formation. Permeabilities of bottom 50 layers are used to create the kernel matrix.

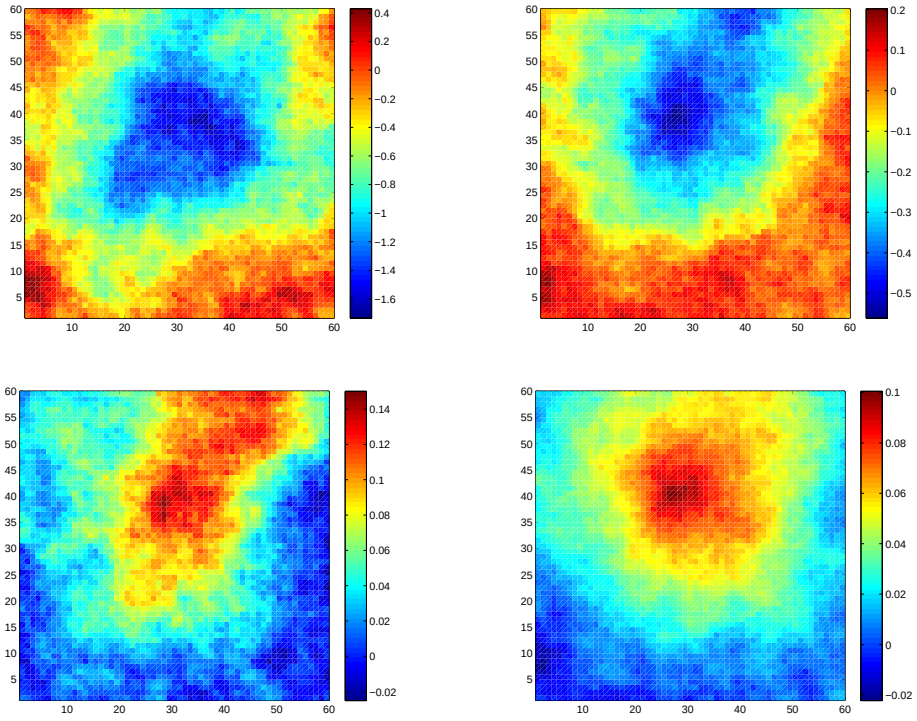


Figure 3.7 : Mean of 10 (top left), 100 (top right), 1000 (left bottom) and 10000 (right bottom) realizations of the natural logarithm of the random permeability field for the porous medium of Tarbert formation by linear PCA.

Ness formation. Fig 3.6 presents four realizations of the random permeability field of Upper Ness formation by linear PCA.

Fig 3.7 and Fig 3.8 present the mean and variance of $10, 100, 1000, 10000$ realizations of the random permeability field for the porous medium of Tarbert formation.

Fig 3.9 and Fig 3.10 present the mean and variance of $10, 100, 1000, 10000$ realizations of random permeability field for porous medium of Upper Ness formation.

Generation of Random Permeability by Kernel PCA

Linear PCA only preserves two-point statistics. Numerically, it appears that kernel PCA performs much better than linear PCA to capture complex geological structures,

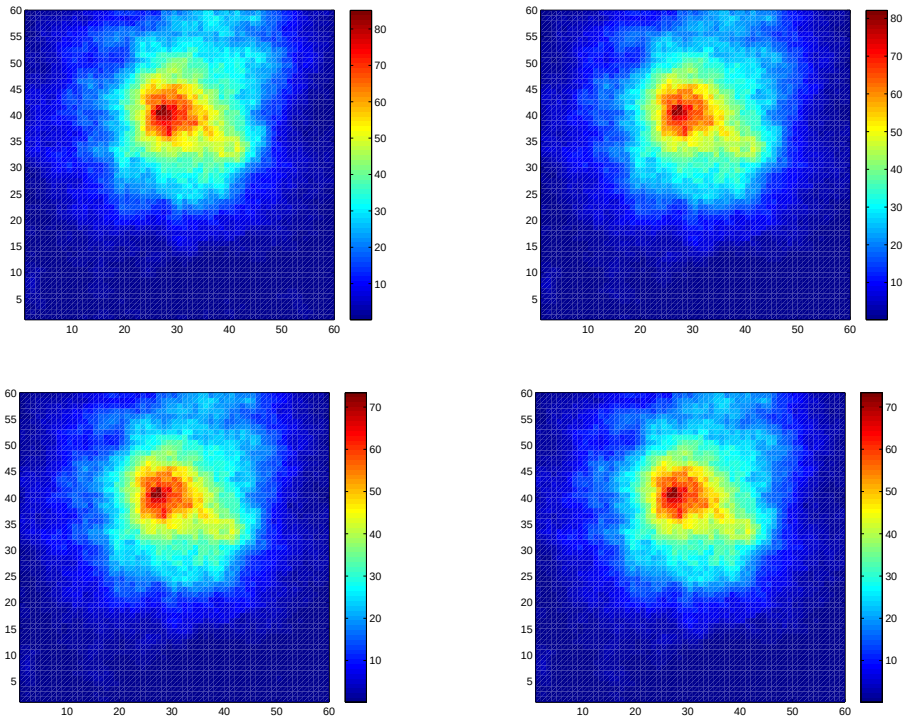


Figure 3.8 : Variance of 10 (top left), 100 (top right), 1000 (left bottom) and 10000 (right bottom) realizations of the natural logarithm of the random permeability field for the porous medium of Tarbert formation by linear PCA.

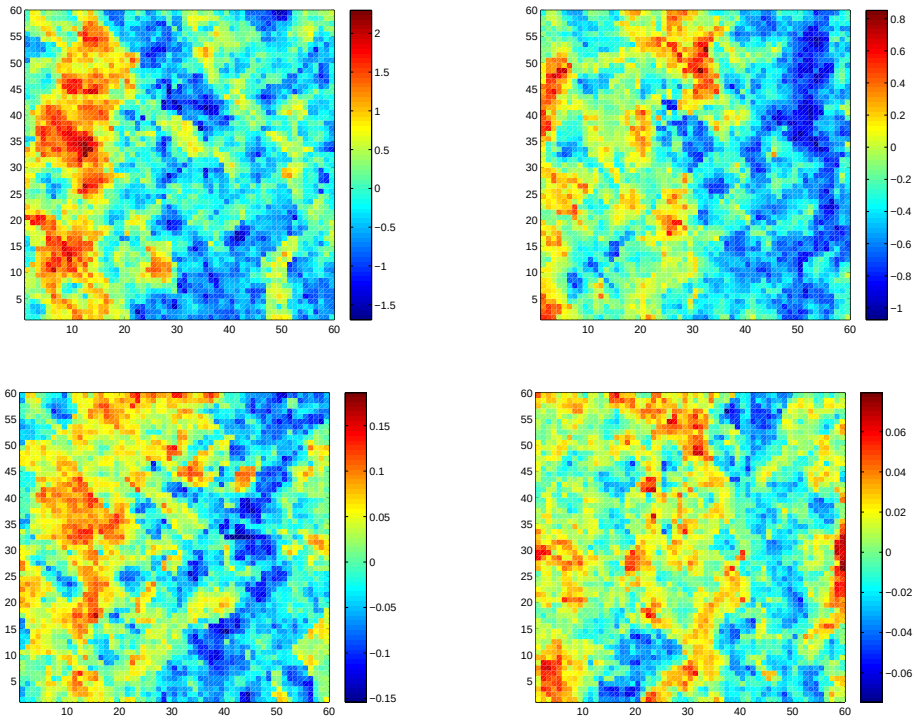


Figure 3.9 : Mean of 10 (top left), 100 (top right), 1000 (left bottom) and 10000 (right bottom) realizations of the natural logarithm of the random permeability field for porous medium of Upper Ness formation by linear PCA.

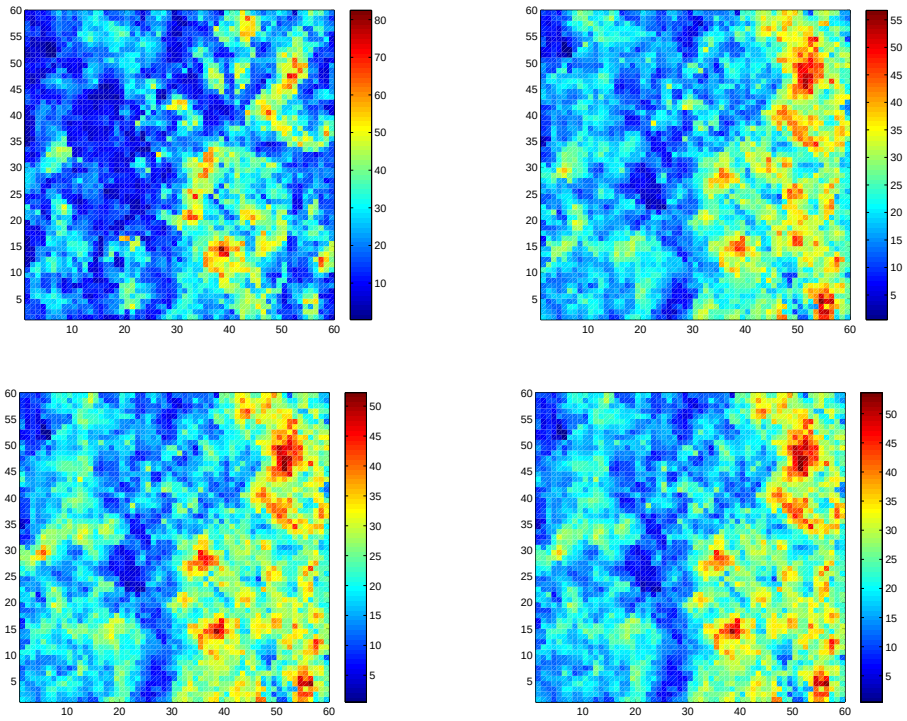


Figure 3.10 : Variance of 10 (top left), 100 (top right), 1000 (left bottom) and 10000 (right bottom) realizations of the natural logarithm of the random permeability field for porous medium of Upper Ness formation by linear PCA.

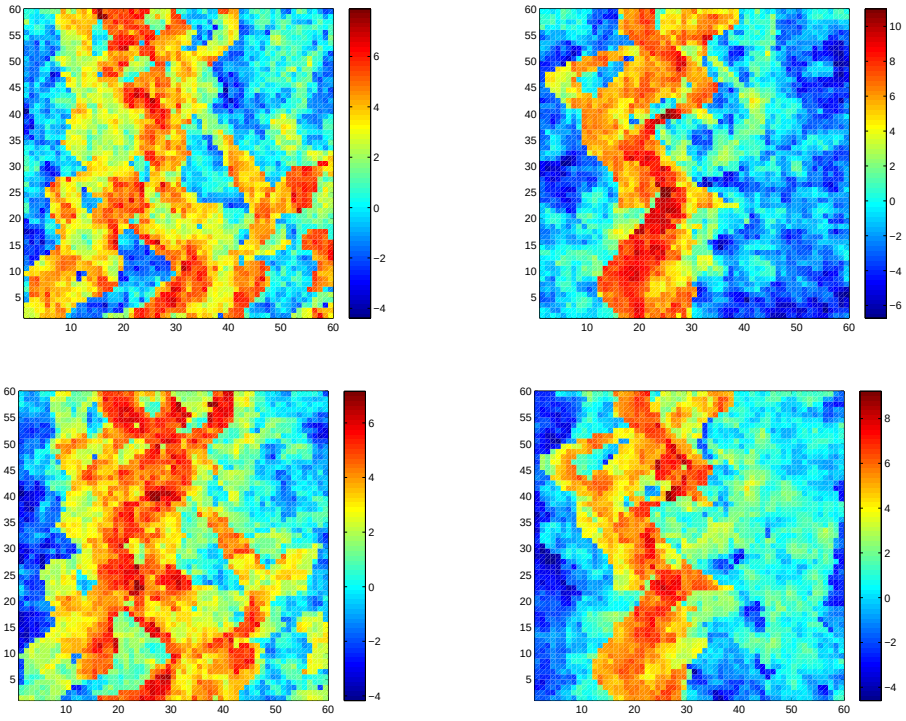


Figure 3.11 : Four realizations of natural logarithm of permeability values for porous medium of Upper Ness formation, which are generated by polynomial kernel PCA of order two.

such as channels. In my thesis, I use polynomial kernel PCA of order two to generate the random permeability field for porous medium of Tarbert formation and Upper Ness formation. The non-linear map is defined in (3.26) and the pre-image iterative scheme is defined in (3.39).

Fig 3.11 shows four realizations generated by polynomial kernel PCA of order two for porous medium of Upper Ness formation. I first apply formula (3.26) on the original permeability realizations and obtain the image of the non-linear map. Then the kernel matrix is created using the images of these realizations.

In Fig 3.11, channels are reproduced successfully, unlike the realizations in Fig 3.5. We are interested in the mean and variance of realizations generated by kernel PCA.

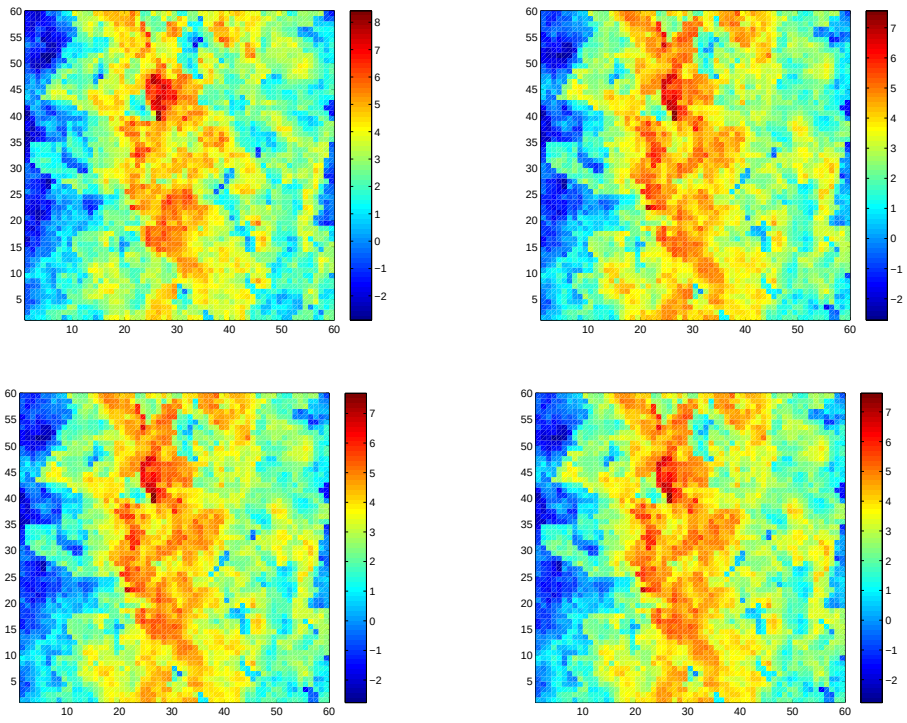


Figure 3.12 : Mean of 10 (top left), 100 (top right), 1000 (left bottom) and 10000 (right bottom) realizations of the natural logarithm of the random permeability field for the porous medium of Upper Ness formation. They are generated by second order polynomial kernel PCA.

Fig 3.12 and Fig 3.13 present the mean and variance of 10, 100, 1000 and 10000 realizations which are generated by second order polynomial kernel PCA, respectively.

By comparing with Fig 3.9, we find that channels appear in Fig 3.12. The maximum and minimum of the permeability values in Fig 3.12 are in the same scale as the original one in Fig 3.2, while the range of permeability values in Fig 3.9 is much smaller. In Fig 3.13, a blue stripe appears as the number of realizations increases. It implies that the variance in that area is smaller than the other areas and the realizations reproduce the channel structure in the blue stripe area. Therefore, the second order polynomial kernel PCA can be used to reproduce the channel structures.

Although no channel structure appears in the porous medium of Tarbert forma-

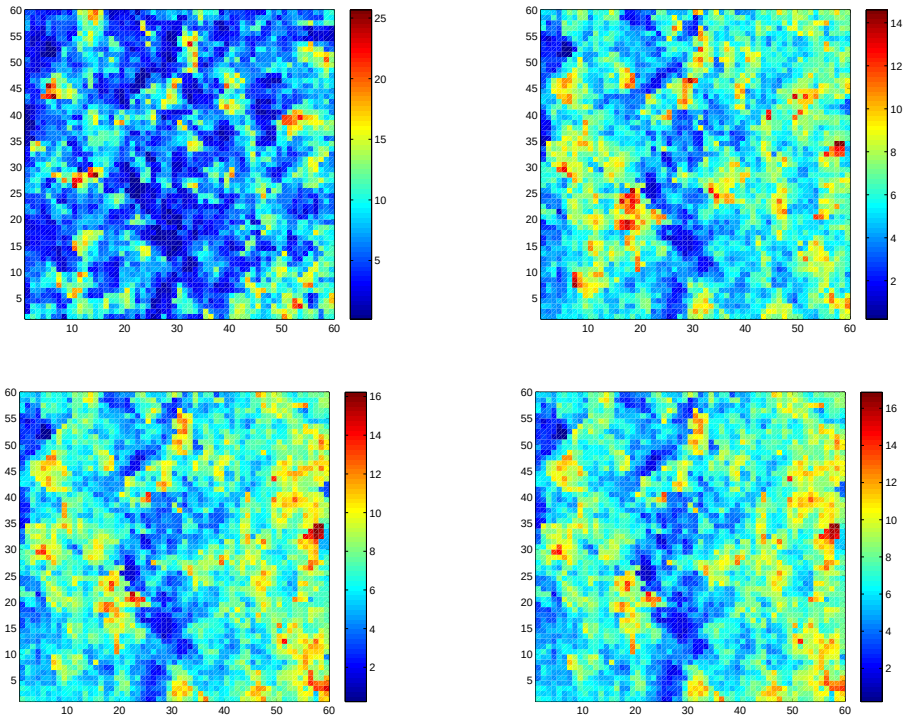


Figure 3.13 : Variance of 10 (top left), 100 (top right), 1000 (left bottom) and 10000 (right bottom) realizations of the natural logarithm of the random permeability field for the porous medium of Upper Ness formation. They are generated by second order polynomial kernel PCA.

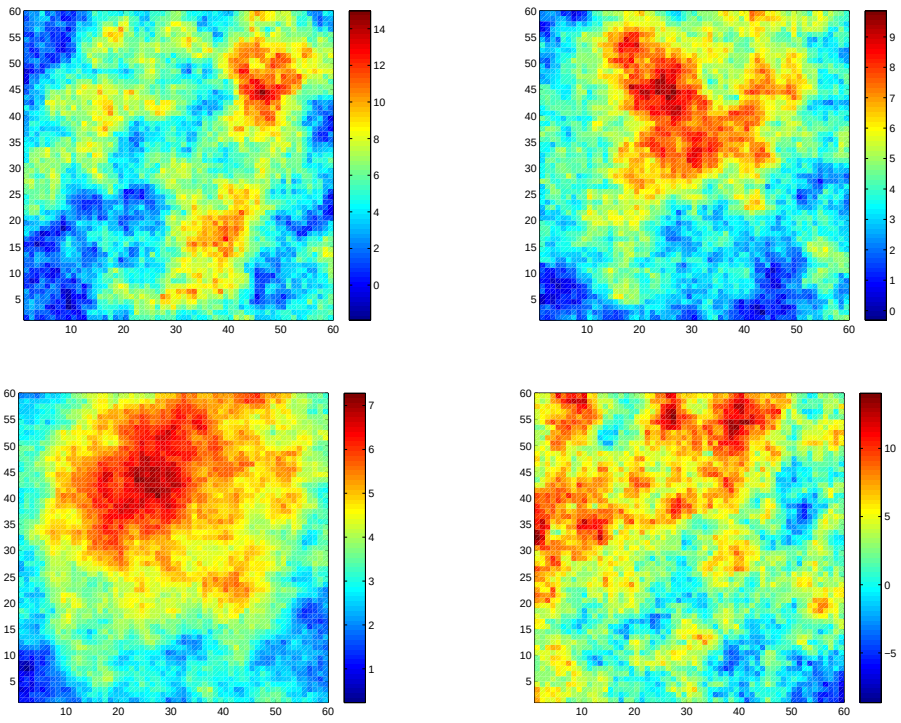


Figure 3.14 : Four realizations of natural logarithm of permeability values for porous medium of Tarbert formation, which are generated by polynomial kernel PCA of order two.

tion, I use kernel PCA to generate the random permeability field for the porous medium of Tarbert formation. Fig 3.14 presents four realizations by second order polynomial kernel PCA method for the porous medium of Tarbert formation.

Fig 3.15 and Fig 3.16 present the mean and variance of 10, 100, 1000 and 10000 simulations generated by second order polynomial kernel function for Top 35 case.

Compared with Fig 3.7, Fig 3.15 shows that the mean of permeability fields generated by kernel PCA is in the same scale as the original one. On the other hand, as shown in Fig 3.16, the variance of realizations generated by kernel PCA increases as the number of simulations increases, while the variance generated by linear PCA stays in the same scale, which is shown in Fig 3.8. As we know, the larger variance,

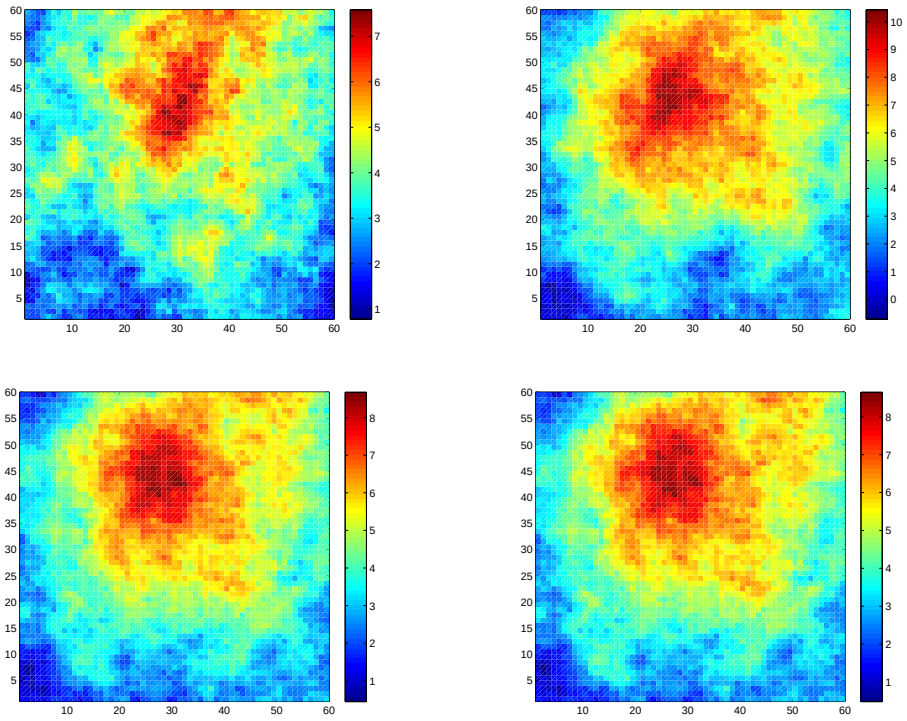


Figure 3.15 : Mean of 10 (top left), 100 (top right), 1000 (left bottom) and 10000 (right bottom) realizations of the natural logarithm of the random permeability field for the porous medium of Tarbert formation. They are generated by second order polynomial kernel PCA.

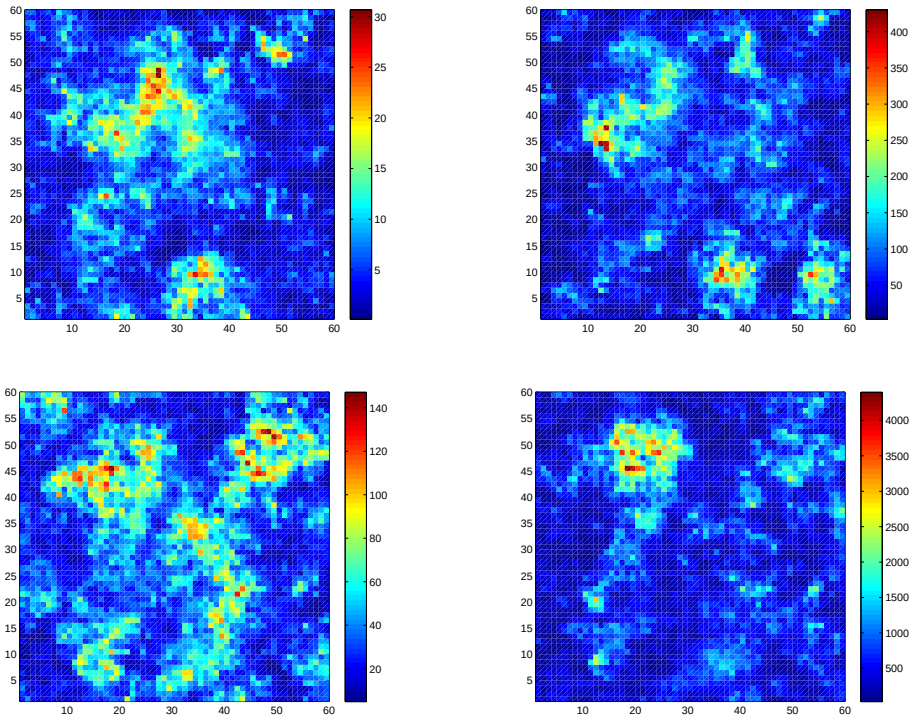


Figure 3.16 : Variance of 10 (top left), 100 (top right), 1000 (left bottom) and 10000 (right bottom) realizations of the natural logarithm of the random permeability field for the porous medium of Tarbert formation. They are generated by second order polynomial kernel PCA.

the more meaningful these realizations are. Furthermore, since kernel PCA is able to capture channel structures, I choose kernel PCA to generate random permeability fields for the porous medium of Tarbert formation and Upper Ness formation.

3.4 Application to Single Phase Flow

In this section, I apply the random permeability fields generated by second order polynomial kernel PCA to a single phase flow model by using MCDG.

The numerical example is:

$$\begin{aligned} -\nabla \cdot (K\nabla u) &= 0 \quad \text{in} \quad (0,360)\times(0,360) \\ u &= 6.2\times 10^9 \quad \text{on} \quad \Lambda_{D1} \\ u &= 5.0\times 10^9 \quad \text{on} \quad \Lambda_{D2} \\ K\nabla u\cdot \mathbf{n} &= 0 \quad \text{on} \quad \Lambda_{N1}\cup\Lambda_{N2}. \end{aligned} \tag{3.46}$$

Fig 3.17 shows the domain for this example. The boundaries Λ_{D1} and Λ_{D2} are Dirichlet boundaries. The boundaries Λ_{N1} and Λ_{N2} are Neumann boundaries.

3.4.1 Porous Medium of Tarbert Formation

In this section, I apply kernel PCA to generate the random permeability field for the porous medium of Tarbert formation and Upper Ness formation. The algorithm is given. The mesh for (3.46) contains 60×60 square cells. The length of each side of the cell is 6 meters. Therefore, the size of the realizations of the random permeability field generated by kernel PCA is 60×60. In order to reduce the computing time and keep certain accuracy, I apply these random permeability realizations of size 60×60 on coarser grids. The sizes of the coarser grids are 15×15 and 30×30. In Fig 3.18, the random permeability realizations of size 15×15 and 30×30 are obtained from

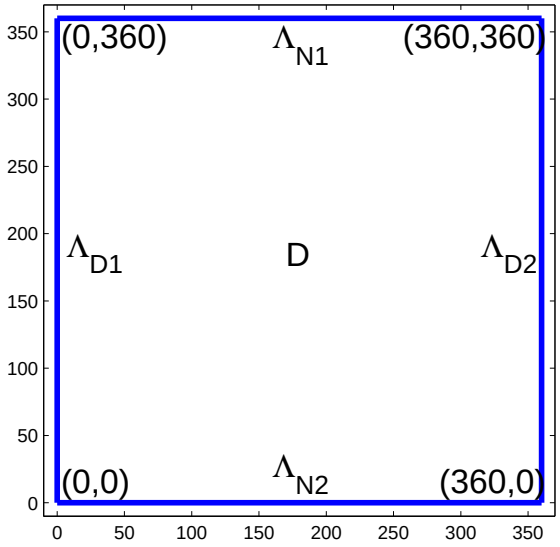


Figure 3.17 : Domain of single phase flow model with Neumann and Dirichlet boundary conditions. $\Lambda_{D1}, \Lambda_{D2}$ are Dirichlet boundaries. $\Lambda_{N1}, \Lambda_{N2}$ are Neumann boundaries

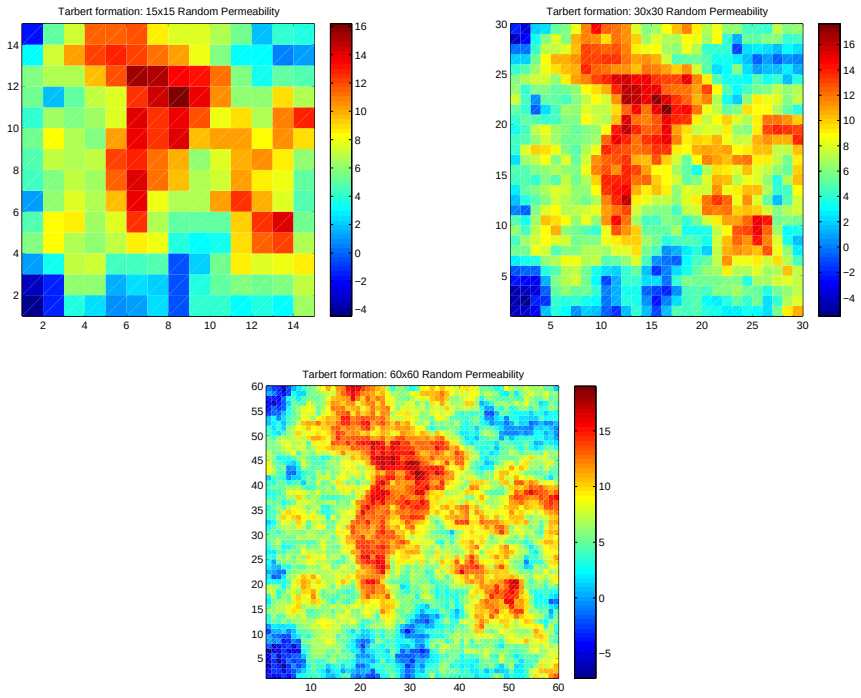


Figure 3.18 : Porous medium of Tarbert formation: random permeability realizations on 15×15 (top left), 30×30 (top right), 60×60 (the second row) grids

the realizations of size 60×60. The approach is by averaging. For example, in order to obtain the permeability of the cell in the left bottom corner on 30×30 grid, I average the permeabilities of the four cells in the left bottom corner of the 60×60 permeability. In the similar way, I average the permeability values of the 16 cells in left bottom corner in 60×60 grid to obtain the permeability value of the cell in the left bottom corner in 30×30 grid. We remark that the input permeability of size 60×60 used to obtain 30×30 and 15×15 are different. Therefore, the permeability on the grid 15×15 is not an average of the permeability on the grid 30×30.

The MCDG method is applied to solve the single phase flow model, defined by (3.46), on 15×15 and 30×30 grids. Polynomial approximation of order two are used and the parameters in the DG method are $\sigma = 1, \epsilon = 1$. Fig 3.19 shows the MCDG solutions with 10000 simulations for the porous medium of Tarbert formation.

The reason that MCDG is applied on coarser grids is that the computing time is largely reduced compared with the time on original 60×60 grid. Fig 3.20 shows the improvement in terms of computing time when MCDG is applied on 15×15, 30×30, 60×60 grids with 10000 simulations. Since the computing time for each simulation is almost the same on the same grid, the computing time of 10000 simulations on 60×60 grid is obtained by multiplying the computing time of 100 simulations by 100.

Fig 3.20 shows that when we apply the problem on coarser grids of size 15×15 and 30×30, the computing time is tremendously reduced compared to the time on 60×60 grid. On the other hand, the solutions on coarser grids are stable and keep a certain accuracy which is shown in Fig 3.19

Fig 3.20 implies that even if we apply MCDG on coarser grid of size 30×30, the computing time of 10000 simulations is about seven and half hours. In order to reduce the computing time furthermore, we use the parallel machine STIC at Rice Research

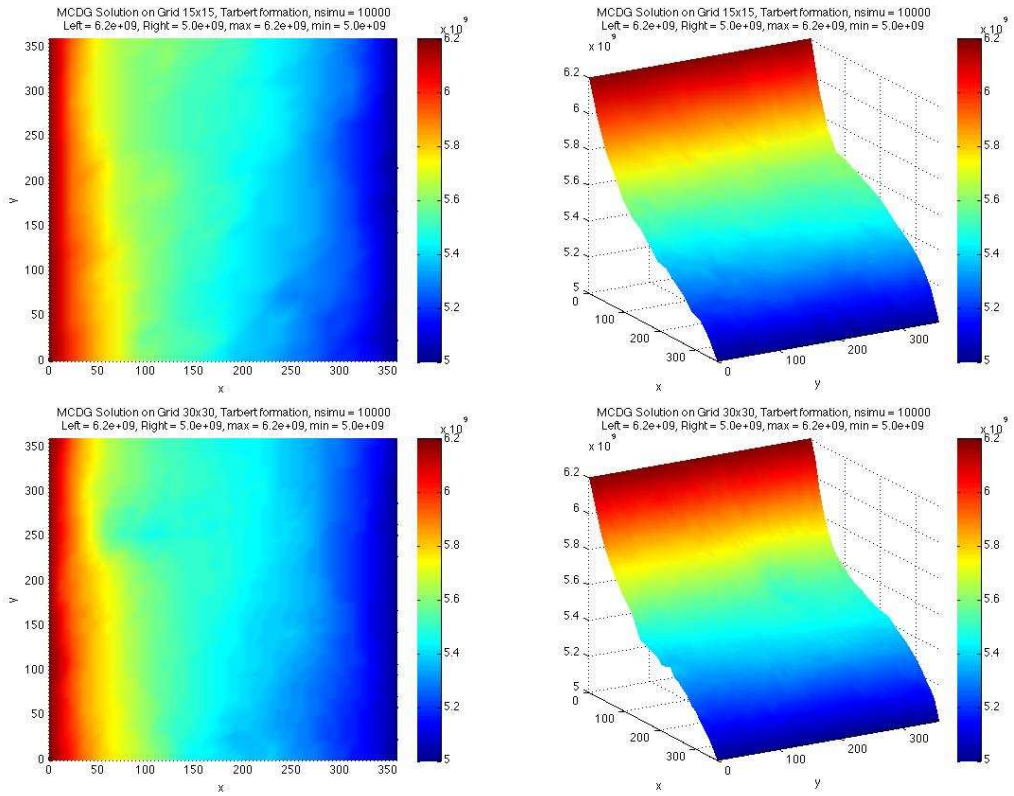


Figure 3.19 : Porous medium of Tarbert formation: MCDG solutions with 10000 simulations on 15×15 (first row), 30×30 (second row) grids in two different view angles. Value on the left boundary is 6.2×10^9. Value on the right boundary is 5.0×10^9. Maximum value is 6.2×10^9, minimum value is 5.0×10^9.

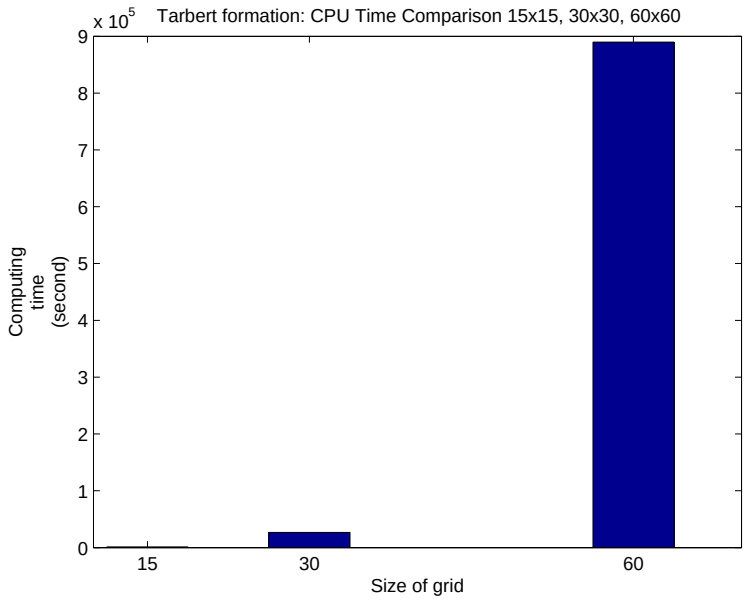


Figure 3.20 : Porous medium of Tarbert formation: computing time of MCDG with 10000 simulations on 15×15, 30×30 and 60×60 grids.

Computing Support Group. Table 3.2 presents the computing time when we use 32, 64 and 128 processors of STIC to obtain MCDG solutions with 10000 simulations on 15×15 and 30×30 grids. It only takes about four minutes to obtain the MCDG solution on 30×30 grid with 10000 simulations.

3.4.2 Porous Medium of Upper Ness Formation

In the porous medium of Tarbert formation, the complex geological structures, such as channels, do not appear. However, in the porous medium of Upper Ness formation, channels appear. In this section, I use kernel PCA to generate the random permeability of the porous medium of Upper Ness formation and apply them to solving the single phase flow model (3.46).

As explained in the previous sections, kernel PCA is applied on a grid of size

Table 3.2 : Porous medium of Tarbert formation: computing time of MCDG with 10000 simulations on STIC with 32, 64 and 128 processors on 15×15, 30×30 grids.

Grid	Number of processors	Computing Time
15×15	128	00 : 00 : 13
15×15	64	00 : 00 : 21
15×15	32	00 : 00 : 40
30×30	128	00 : 03 : 59
30×30	64	00 : 07 : 01
30×30	32	00 : 13 : 52

60×60. Then we obtain averaged permeabilities on coarser grids of size 15×15 and 30×30. Fig 3.21 presents one realization of the random permeability field generated by kernel PCA for the porous medium of Tarbert formation. In Fig 3.21, channels are reproduced successfully.

Fig 3.22 presents the MCDG solutions with 10000 simulations on coarser grid of size 15×15 and 30×30 from two different view angles. Fig 3.22 implies that the MCDG solutions satisfy well the boundary conditions. Polynomial approximations of order two are used and the parameters in the DG method are $\sigma = 1, \epsilon = 1$.

Fig 3.23 shows the computing time of MCDG on grids of size 15×15, 30×30, 60×60 with 10000 simulations. Fig 3.23 indicates that the computing time of MCDG is tremendously reduced on grids of size 15×15 and 30×30 compared with the computing time on grid of size 60×60. Meanwhile, the MCDG solution is stable and the shape of the solution is maintained.

Fig 3.20 also shows that even if we apply MCDG on coarser grid of size 30×30,

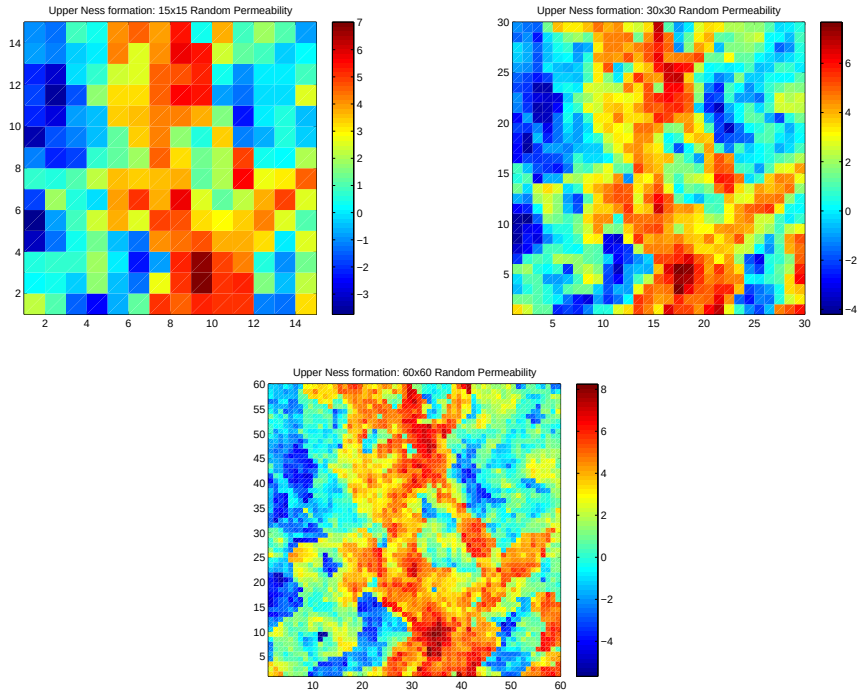


Figure 3.21 : Porous medium of Upper Ness formation: one permeability realization on coarser grids of size 15×15 (top left), 30×30 (top right), 60×60 (the second row) by kernel PCA.

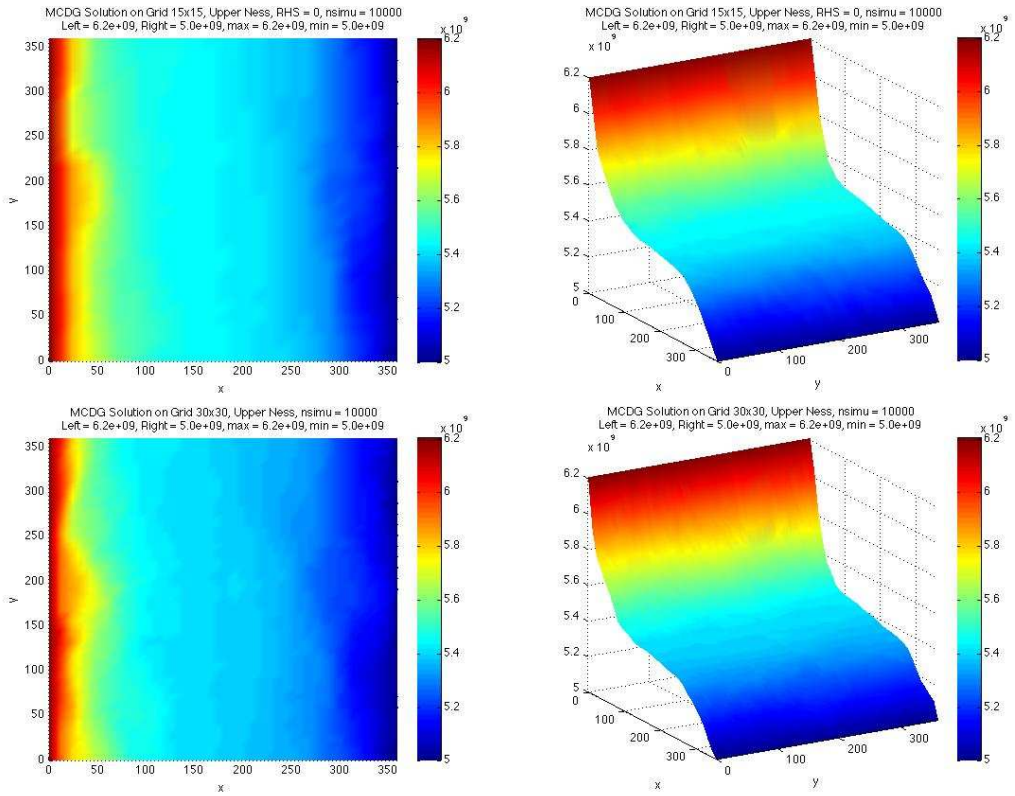


Figure 3.22 : Porous medium of Upper Ness formation: MCDG solutions with 10000 simulations on 15×15 (first row) and 30×30 (second row) grids in two different view angles. Value on the left boundary is 6.2×10^9. Value on the right boundary is 5.0×10^9. Maximum value is 6.2×10^9, minimum value is 5.0×10^9.

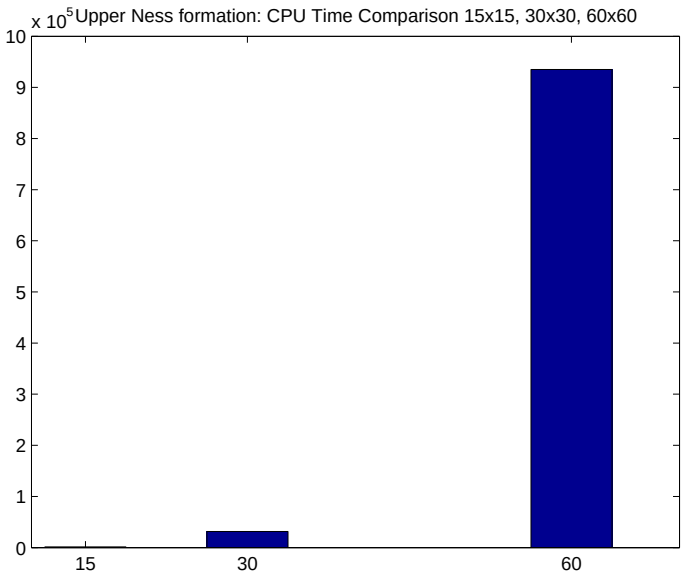


Figure 3.23 : Porous medium of Upper Ness formation: Computing time of MCDG with 10000 simulations on grids of size 15×15, 30×30 and 60×60.

the computing time of solving 10000 simulations is about nine hours. In order to reduce the computing time, I use STIC to address this problem on 32, 64 and 128 processors. Table 3.3 presents the computing time of MCDG on 32, 64 and 128 processors of STIC with 10000 simulations on grids of size 15×15 and 30×30. It only takes less than five minutes to obtain the MCDG solution on 30×30 grid with 10000 simulations.

Table 3.3 : Porous medium of Upper Ness formation: computing time of MCDG with 10000 simulations on STIC with 32, 64 and 128 processors on girds of size 15×15, 30×30.

Grid	Number of processors	Computing Time
15×15	128	00 : 00 : 15
15×15	64	00 : 00 : 24
15×15	32	00 : 00 : 44
30×30	128	00 : 04 : 18
30×30	64	00 : 08 : 18
30×30	32	00 : 16 : 35

Chapter 4

Coupled Flow and Transport

4.1 Introduction

This chapter studies the coupled flow and transport problem with uncertain coefficients. The pressure equation is first solved; and this yields a velocity vector that is used as input data for the transport problem. The transport problem is essentially a parabolic partial differential equation with a convection term. In this chapter, I discuss the case where the permeability coefficient in the pressure equation is a random field, which leads the convection term in the transport problem to be a random field. In the numerical example, the permeability value randomly oscillates between the range of $[\mathcal{O}(10^{-8}), \mathcal{O}(10^{4})]$ millidarcy. Due to this high level of uncertainty and oscillation, the overshoot and undershoot phenomena appear in the approximation to the solution to the transport equation. I use slope limiter and set upper and lower bounds constraints to address this difficulty. Furthermore, I develop one approach to approximate the solution to this coupled problem and compare the results in two types of heterogeneous porous media.

This chapter is organized as follows. In Section 1, I present the model problem with certain boundary and initial conditions. In Section 2, one approach is developed and presented to solve the coupled flow and transport problem. In Section 3, a benchmark example, which contains two different cases, is created to show the effectiveness of this approach. Conclusions are obtained by comparing the results of two types of

heterogeneous porous media.

4.2 Model Problem

Let $(\Omega, \mathcal{F}, P)$ be a complete probability space, where Ω is the outcome space, $\mathcal{F}$ is the $\sigma-$algebra of events, and $P : \mathcal{F} \to [0,1]$ is a probability measure. Assume $D \subset \mathbb{R}^d$ is a convex bounded domain. The boundary of the domain ∂D is assumed to be the union of two disjoint sets Λ_D (Dirichlet boundary) and Λ_N (Neumann boundary).

Flow Equation: It is an elliptic partial differential equations with random coefficients. We want to find a stochastic function, $u : \Omega \times \bar{D} \to \mathbb{R}$, such that the following equations are $P-$a.e. satisfied:

$$\begin{aligned} \nabla \cdot (K_a(w,\cdot)\nabla u(w,\cdot)) = 0 \quad \text{in} \quad D, \\ u(w,\cdot) = u_D(w,\cdot) \quad \text{on} \quad \Lambda_D, \\ -(K_a(w,\cdot)\nabla u(w,\cdot)) \cdot \mathbf{n} = u_N(w,\cdot) \quad \text{on} \quad \Lambda_N. \end{aligned} \tag{4.1}$$

where $K_a, u_D, u_N : \Omega \times D \to \mathbb{R}$ are stochastic functions with continuous and bounded covariance functions. The solution u denotes the pressure of the flow, and K_a denotes the permeability of the porous media. Additionally, I assume that K_a is bounded, uniformly coercive and the first derivative of K_a is uniformly bounded and continuous. The vector $\mathbf{n}$ denotes the outward unit normal vector to the boundary ∂D.

Using the finite dimensional noise assumption mentioned in Chapter 2, the model problem (4.1) can be transformed into a parametrized problem as below:

$$\begin{aligned} -\nabla \cdot K_a(y,x)\nabla u(y,x) = 0 \quad \forall (y,x) \in \Gamma \times D, \\ u(y,x) = u_D(y,x) \quad \forall (y,x) \in \Gamma \times \Lambda_D, \\ -(K_a(y,x)\nabla u(y,x)) \cdot \mathbf{n} = u_N(y,x) \quad \forall (y,x) \in \Gamma \times \Lambda_N. \end{aligned} \tag{4.2}$$

where $\Gamma \subset \mathbb{R}^N$ is the support of the joint probability, which is defined in section 2.2.2.

Transport Equation: I discuss the transport equation with random convection term, which is obtained from solving the flow equation (4.1). I want to find a stochastic function $c : [0,T] \times \Omega \times \bar{D}$ such that the following equations are $P-$a.e. satisfied:

$$\begin{aligned}
\frac{\partial c(w,\cdot)}{\partial t} - \nabla \cdot (K_b \nabla c(w,\cdot) - \boldsymbol{\beta}(u(w,\cdot))c(w,\cdot)) = 0 \quad \text{in} \quad (0,T] \times D, \\
c(w,\cdot) = c_D(w,\cdot) \quad \text{on} \quad (0,T] \times \Lambda_D, \\
-(K_b \nabla c(w,\cdot)) \cdot \mathbf{n} = c_N(w,\cdot) \quad \text{on} \quad (0,T] \times \Lambda_N, \\
c((w,\cdot)) = c_0(w,\cdot) \quad \text{on} \quad \{0\} \times D,
\end{aligned} \tag{4.3}$$

where $\boldsymbol{\beta}$ is the random convection term, which is a function of the pressure u, and K_b is a deterministic function. In fact, the relationship between u and $\boldsymbol{\beta}$ is given by Darcy's law:

$$\boldsymbol{\beta}(u) = -K_a \nabla u. \tag{4.4}$$

With the finite dimensional noise assumption, we transform the equation (4.3) into a parametrized problem:

$$\begin{aligned}
\frac{\partial c(t,y,x)}{\partial t} - \nabla \cdot (K_b \nabla c(t,y,x) - \boldsymbol{\beta}(u(y,x))c(t,y,x)) \\
= 0 \quad \forall (t,y,x) \in (0,T] \times \Gamma \times D, \\
c(t,y,x) = c_D(t,y,x) \quad \forall (t,y,x) \in (0,T] \times \Gamma \times \Lambda_D, \\
-(K_b \nabla c(t,y,x)) \cdot \mathbf{n} = c_N(t,y,x) \quad \forall (t,y,x) \in (0,T] \times \Gamma \times \Lambda_N, \\
c(0,y,x) = c_0(y,x) \quad \forall (t,y,x) \in \{0\} \times \Gamma \times D.
\end{aligned} \tag{4.5}$$

The solution to the coupled system (4.2), (4.5) can be approximated by combining sampling techniques with locally mass conservative methods. In this chapter, I use the Monte Carlo discontinuous Galerkin method to solve this parametrized coupled problem.

4.3 Monte Carlo Discontinuous Galerkin Scheme

In this section, I propose one approach to approximate the solution to the coupled flow and transport system defined in section 4.2. Since the convection term is a function of the solution to the flow equation with random permeability coefficient, it is a random field. I take the uncertainty in the coupled system into consideration by using Monte Carlo sampling technique. In this approach, the flow equation is solve by MCDG and the resulting velocity obtained from each simulation becomes input data to the transport problem. In that case, the transport problem is stochastic.

In this approach, the MCDG method is applied to take the uncertainty of flow equation (4.2) into consideration. I generate M samples of velocity fields by solving (4.2) with MCDG. Then, for each sample of velocity field, I solve the transport equation. Therefore, I solve M transport problems. At the end, I average all M solutions to the transport equation to obtain the MCDG solution to the coupled system. The algorithm is given as follows:

1. **Step one: Choose DG space and number of realizations**

   - Choose a number of realizations, $M \in \mathbb{N}_+$, and a DG space $D_r(\mathcal{E}_h)$, with $r \in \{r_1, r_2\}$.

2. **Step two: Solve coupled system**

   - For $i = 1, \cdots, M$
     - **Solve flow equation**
       * Use kernel PCA to generate random permeability field K_a^i.
       * Solve $A_\epsilon(K_a^i; u_h^i, v) = L(v)$, $\forall v \in D_{r_1}(\mathcal{E}_h)$.
       * Compute $\boldsymbol{\beta}^i(K_a^i, u_h^i) = -K_a^i \nabla u_h^i$.

where we recall the bilinear form defined in section 2.3.2:

$$\begin{aligned} A_\epsilon(K_a;u,v) &= \sum_{E\in\mathcal{E}_h}\int_E K_a\nabla u\cdot\nabla v - \sum_{e\in D_h\cup\Lambda_D}\int_e \{K_a\nabla u\cdot\mathbf{n}_e\}\,[v] \\ &\quad + \epsilon\sum_{e\in D_h\cup\Lambda_D}\int_e \{K_a\nabla v\cdot\mathbf{n}_e\}\,[u] + \sum_{e\in D_h\cup\Lambda_D}\frac{\sigma}{h}\int_e [v][w], \\ L(v) &= \epsilon\sum_{e\in\Lambda_D}\int_e \Big(K_a\nabla v\cdot\mathbf{n}_e + \frac{\sigma}{h}v\Big)\,u_D + \sum_{e\in\Lambda_N}\int_e v u_N. \end{aligned} \tag{4.6}$$

– **Solve transport equation:** $\forall v \in D_{r_2}(\mathcal{E}_h)$, find $(c_h^n)_{n=0}^{N_t} \in D_{r_2}(\mathcal{E}_h)$, such that

$$\begin{aligned} \left(\frac{c_h^{m+1}-c_h^m}{\Delta t}, v\right)_{L^2(D)} &+ A_\epsilon(K_b; c_h^{m+1}, v) + B(\boldsymbol{\beta}^i; c_h^{m+1}, v) \\ &= L(v), \quad m = 0, 1, \cdots, N_t - 1 \\ (c_h^0, v)_{L^2(D)} &= (c_0, v)_{L^2(D)}, \end{aligned} \tag{4.7}$$

where N_t is the number of time steps, A_ϵ and $L(v)$ are defined in (4.6) with u_D and u_N replaced by c_D and c_N, B is defined as

$$\begin{aligned} B(\boldsymbol{\beta}(u);u,v) = -\sum_{E\in\mathcal{E}_h}\int_E u\boldsymbol{\beta}(u)\cdot\nabla v + \sum_{e\in D_h}\int_e \{\boldsymbol{\beta}(u)\cdot\mathbf{n}_e\}\,u^{up}[v] \\ + \sum_{e\in\Lambda_D\cup\Lambda_N}\int_e \boldsymbol{\beta}(u)\cdot\mathbf{n}_e uv \end{aligned} \tag{4.8}$$

– **Obtain the solution to the ith simulation**

$$c_h^{N_t}(\boldsymbol{\beta}^i,\cdot) = c_h^{N_t}(\cdot)$$

- End

3. **Approximate the expected value of the solution at time T**

$$\mathbb{E}[c_h] = \frac{1}{M}\sum_{i=1}^{M} c_h^{N_t}(\boldsymbol{\beta}^i,\cdot). \tag{4.9}$$

In the following section, I create one numerical example with benchmarking data to show the effectiveness of this approach.

4.4 Numerical Example

In this section, I apply the approach proposed in section 4.3 to solve a benchmark problem. The permeability values of the SPE10 model are used to generate random permeability fields by kernel PCA for a porous medium of Tarbert formation and a porous medium of Upper Ness formation (see section 3.3.4). The realizations of the permeability field are used to generate the samples of random velocity field from the pressure equation.

The large range of the permeability values leads to significant variations in the velocity field. Therefore, possible overshoot occurs when we solve the transport equation. In this section, I first introduce a slope limiter which can greatly reduce the overshoot and undershoot phenomena. Then, I present the benchmark problem and the numerical solutions.

4.4.1 Slope Limiters

Limiting procedure [49–51] has been widely applied to avoid the spurious oscillations of numerical solutions while preserving the high-order accuracy of DG methods. In this section, I describe a simple slope limiter for two dimensional triangular elements [27]. I assume that the numerical solution c_h to the transport problem is piecewise linear and its restriction to an element E can be written as:

$$c_h(x, y) = a_0 + a_1 x + a_2 y, \quad \forall (x, y) \in E.$$

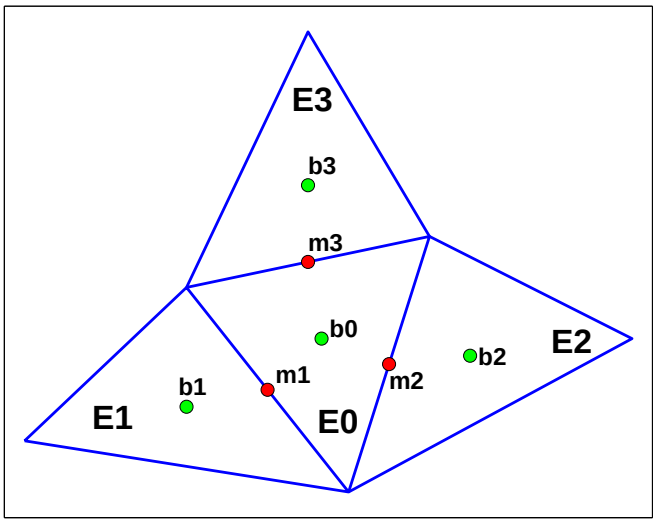


Figure 4.1 : Triangle configuration for building the slope limiter, where m_1, m_2, m_3 are the midpoints of the corresponding edges, b_1, b_2, b_3 are the gravity centers of the elements $E1, E2, E3$.

The slope limiter adjusts the slope of $c_h|_E$ to reduce the overshoot or undershoot phenomena, while conserving the average.

There are four steps to construct the slope limiter:

1. **Step one: compute neighbour average:** I first compute the average concentration for the element to be limited and all neighbouring elements as follows. In Fig 4.1, assume that a given element E_0 has three neighbours E_1, E_2, E_3. Let $\bar{c}_i$ denote the average solution of c_h over E_i:

$$\bar{c}_i = \frac{1}{|E_i|} \int_{E_i} c_h, \quad i = 0, \cdots, 3. \tag{4.10}$$

2. **Step two: test if limiting is necessary:** Let m_j denote the midpoints of the edges of E_0. Compute the concentration $c_h|_{E_0}(m_j)$ and check if this value is between $\bar{c}_i$ and $\bar{c}_0$. Stop here if the test is successful; otherwise, continue to

step three.

3. **Step three: construct and rank three linears:** Construct three linears using the gravity center b_j of E_j and the averages $\bar{c}_i$. For instance, if we write the linears as $L_j(x, y) = a_0^j + a_1^j x + a_2^j y$, where $j \in \{1, 2, 3\}$, they are uniquely determined by

$$L_j(b_0) = \bar{c}_0 \quad \text{and} \quad L_j(b_k) = \bar{c}_k \quad k \neq j. \tag{4.11}$$

Then, rank the linears by decreasing $\sqrt{(a_1^j)^2 + (a_2^j)^2}$.

4. **Step four: select appropriate linear:** Finally, check if the values of the linears evaluated at the midpoint $m_k, L_j(m_k)$, are between $\bar{c}_k$ and $\bar{c}_0$ for $k = 1, 2, 3$. The linears are tested in the order given in step three. If one of the linears passes this test, it is chosen to be the limited solution. Otherwise, if none of the constructed linears satisfies the test, set the slope to be zero.

4.4.2 Benchmark Problem

Darcy's law [52] forms the relationship of fluid pressure and flow rate through the medium. It also builds the bridge between the flow equation and transport equation which are discussed in this chapter. Darcy's law can be written as

$$\boldsymbol{\beta} = -\frac{K}{\mu}\nabla u \tag{4.12}$$

where

- $\boldsymbol{\beta}$: the superficial fluid flow rate through the medium. Unit is length per time i.e., $m \cdot s^{-1}$.
- K: the permeability of the medium. Unit of permeability is millidarcy (mD), where $1mD = 10^{-15}m^2$.

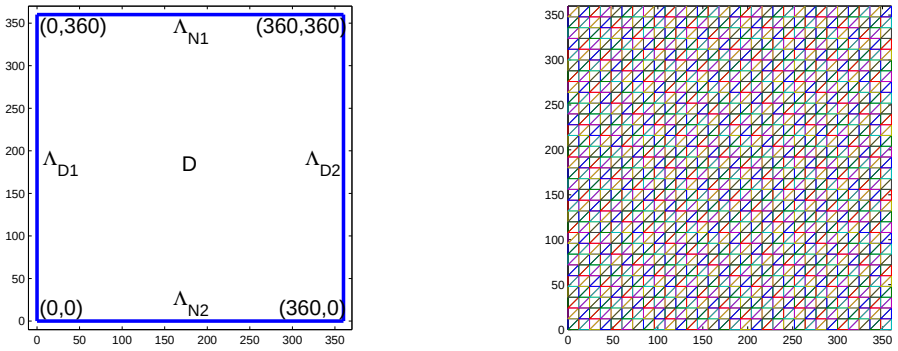


Figure 4.2 : The left figure is the computational domain D (left), where $\Lambda_{D1}, \Lambda_{D2}$ are Dirichlet boundaries and $\Lambda_{N1}, \Lambda_{N2}$ are Neumann boundaries. The right figure is an example of triangular structured mesh.

- μ: the viscosity of the fluid. Unit is cp, i.e., $10^{-3} Pa \cdot s$.
- ∇u: the gradient of pressure. Unit is $Pa \cdot m^{-1}$.

Denoting $K_a = \frac{K}{\mu}$, we have (as in our model problem):

$$\boldsymbol{\beta} = -K_a \nabla u. \tag{4.13}$$

The viscosity of water at 20°C is $1cp$. The unit of permeability provided in the SPE10 model is mD, which equals to $10^{-15} m^2$. Therefore, I multiply the permeability values by $\frac{10^{-15}}{10^{-3}} = 10^{-12}$ to make the unit of fluid flow rate $\boldsymbol{\beta}$ to be ms^{-1}. On the other hand, the pressure in earth increases as the depth increases [53]. I consider two types of rock formation, as described in Section 3.3.1: Tarbert formation and Upper Ness formation.

The computational domain is $(0, 360) \times (0, 360)$. The mesh is a structured mesh consisting of triangular elements. The domain and an example of a mesh are shown in Fig 4.2. The boundary conditions for (4.2) and (4.5) are:

- **Porous Medium of Tarbert Formation:**

$$\begin{aligned} u &= 6.2 \times 10^9 \quad \text{on} \quad \Lambda_{D1}, \\ u &= 5.0 \times 10^9 \quad \text{on} \quad \Lambda_{D2}, \\ K_a \nabla u \cdot \mathbf{n} &= 0 \quad \text{on} \quad \Lambda_{N1} \cup \Lambda_{N2}, \\ c_D &= c_N = 0. \end{aligned} \tag{4.14}$$

- **Porous Medium of Upper Ness Formation:**

$$\begin{aligned} u &= 6.2 \times 10^{10} \quad \text{on} \quad \Lambda_{D1}, \\ u &= 5.0 \times 10^{10} \quad \text{on} \quad \Lambda_{D2}, \\ K_a \nabla u \cdot \mathbf{n} &= 0 \quad \text{on} \quad \Lambda_{N1} \cup \Lambda_{N2}, \\ c_D &= c_N = 0. \end{aligned} \tag{4.15}$$

In the transport equation (4.5), the coefficient K_b represents the molecular diffusion of the fluid. In [54], Table 1 presents the diffusion coefficient of water at different temperatures. At 20°C, the molecular diffusion coefficient K_b is around $2.023 \times 10^{-9} m^2 s^{-1}$.

Note that the unit of time in transport equation (4.5) is the second. In order to observe the simulation results in unit day, I use the change of variable $\tau = \frac{t}{m}$, where $m = 24 \times 3600$. The unit of τ is day and the transport equation (4.5) is transformed into

$$\frac{\partial c}{\partial \tau} - \nabla \cdot (\tilde{K}_b \nabla c - \tilde{\boldsymbol{\beta}}(u)c) = 0 \quad \text{where} \quad \tilde{K}_b = mK_b, \quad \tilde{\boldsymbol{\beta}}(u) = m\boldsymbol{\beta}(u). \tag{4.16}$$

Therefore, the molecular diffusion coefficient

$$\tilde{K}_b = 24 \times 3600 \times 2.023 \times 10^{-9} \approx 2.0 \times 10^{-4}.$$

To summarize, we obtain the following transport problem:

$$\begin{aligned} \frac{\partial c}{\partial \tau} - \nabla \cdot (\tilde{K}_b \nabla c - \tilde{\boldsymbol{\beta}}(u)c) = 0 \quad &\text{in} \quad (0,T) \times D, \\ c = 0 \quad &\text{on} \quad (0,T) \times \{\Lambda_{D1} \cup \Lambda_{D2}\}, \\ -(\tilde{K}_b \nabla c) \cdot \mathbf{n} = 0 \quad &\text{on} \quad (0,T) \times \{\Lambda_{N1} \cup \Lambda_{N2}\}, \\ c(0,\mathbf{x}) = c_0 \quad &\text{on} \quad \{0\} \times D. \end{aligned} \tag{4.17}$$

The initial solution is defined as follows

$$c_0 = \begin{cases} 1.0 & \text{in} \quad D_I, \\ 0 & \text{elsewhere}, \end{cases}$$

where the domain D_I is a small region of the computational domain.

4.4.3 Porous Medium of Tarbert Formation

In this section, I generate 10000 random permeability samples for a porous medium of Tarbert formation, as described in section 3.3. We then obtain 10000 numerical realizations of the random velocity field. I apply the approach proposed in section 4.3 to this type of porous medium.

The initial solution c_0 is defined as follows:

$$c_0 = \begin{cases} 1.0 & (x,y) \in [72,96] \times [264,288], \\ 0 & \text{elsewhere}, \end{cases}$$

which is shown in Fig 4.3.

The reason that we choose the initial solution to be located in $[72,96] \times [264,288]$ is because the average flow velocity is larger in that region than in most of other regions in the domain. This is shown in Fig 4.4. Therefore, the contaminant moves faster in that region and can be observed more apparently than in other parts of the domain.

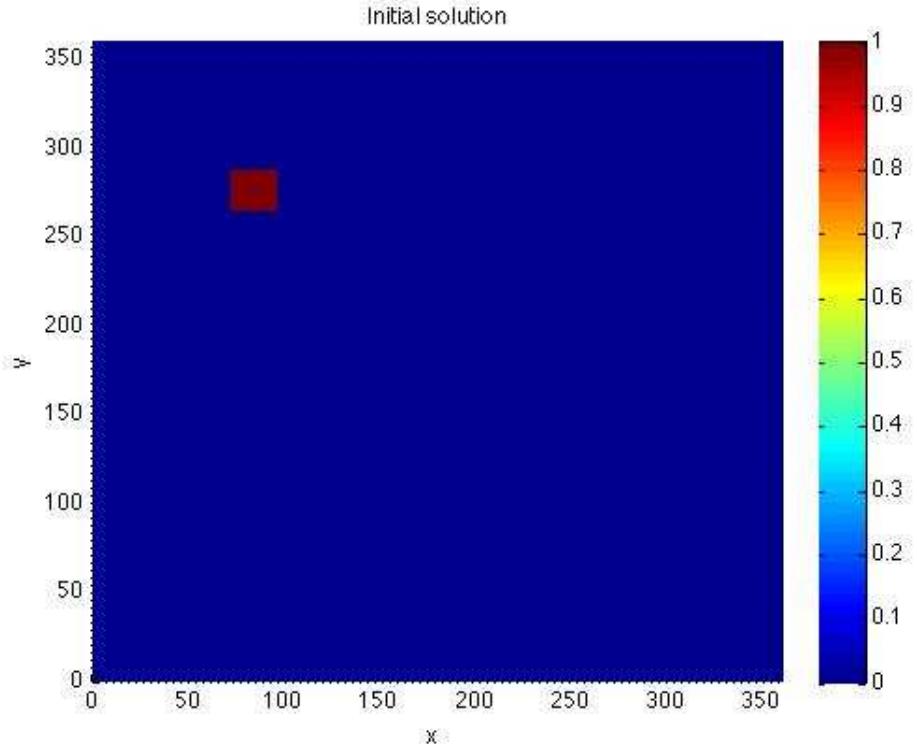


Figure 4.3 : Initial solution for porous medium of Tarbert formation, where $D_I = \{(x, y)|(x, y) \in [72, 96] \times [264, 288]\}$.

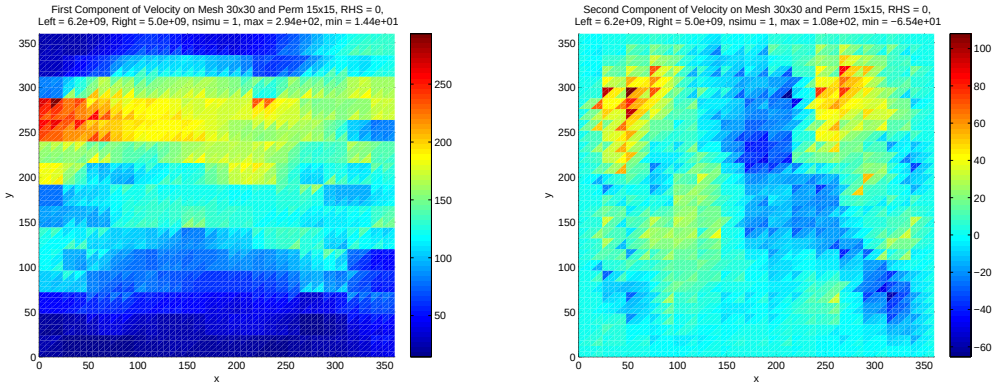


Figure 4.4 : First and second component of the average fluid velocity of 10000 simulations generated in pressure equation with boundary conditions (4.14) by using MCDG for porous medium of Tarbert formation.

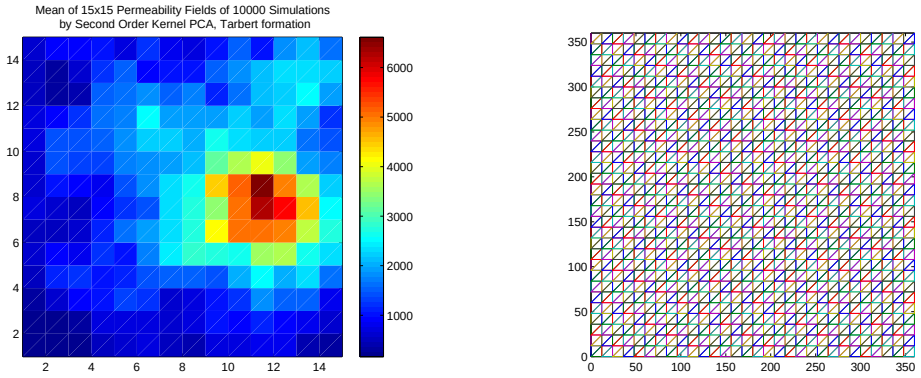


Figure 4.5 : Average of 10000 15×15 permeability fields generated by kernel PCA (left) and the mesh for pressure and flow equations (right) for the porous medium of Tarbert formation. The size of the mesh is 30×30.

The pressure on the boundary indicates that the direction of the transport in this benchmark problem is from left to right. The direction is driven by the boundary conditions (4.14), where the pressure on the left boundary is 6.2×10^9 Pa, while the pressure on the right boundary is 5.0×10^9 Pa.

The value of permeability which is used to generate random permeability fields in my thesis ranges from $\mathcal{O}(10^{-8})$ to $\mathcal{O}(10^4)$ mD. The large range leads to significant oscillation and uncertainty in permeability realizations by kernel PCA, thus, causing a number of solutions with overshoot or undershoot. In order to reduce the impact caused by the oscillations in permeability realizations, I apply the random permeability field of size 15×15 on a finer mesh of size 30×30. Fig 4.5 shows the average of the 10000 permeability realizations of size 15×15 and the mesh for this coupled flow and transport system.

By using these M realizations of random permeability field, the MCDG solution to the flow equation with boundary conditions (4.14) is computed and given in Fig 4.6.

Solving Transport Problem with MCDG

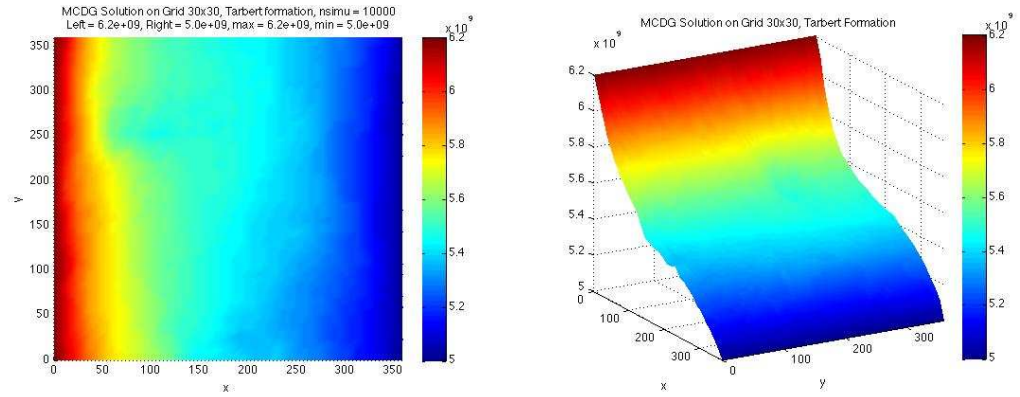


Figure 4.6 : Porous medium of Tarbert formation: MCDG Solution to the flow equation with boundary conditions (4.14) with 10000 simulations in different views. Value on the left boundary 6.2×10^9, right boundary 5.0×10^9, maximum value 6.2×10^9, minimum value 5.0×10^9. The size of permeability is 15×15, while the size of the mesh is 30×30.

In this approach, I apply the 10000 flow velocity realizations, individually, to the transport equation with boundary conditions given in (4.14). It generates 10000 solutions. Because of the high oscillation in each of the velocity samples, the overshoot or undershoot phenomena may appear among the 10000 solutions. I observe that the slope limiter is an effective technique to limit the solution. The example in Fig 4.7 shows that the solution is corrected by slope limiter. In that figure, the left picture is obtained without using slope limiter. We can observe an obvious blow-up element in the upper left of the concentration. After slope limiter is applied, the blow-up phenomena in that element is removed completely and the solution remains bounded.

However, I observe that some of the solutions among 10000 simulations do not preserve positivity, that is because DG methods do not satisfy maximum principle and therefore do not preserve the positivity of the solution to the transport equation. This will be shown later in this chapter.

I also observe that slope limiter cannot fix all of the overshoot or undershoot solutions among 10000 simulations. That is because the range of the permeability

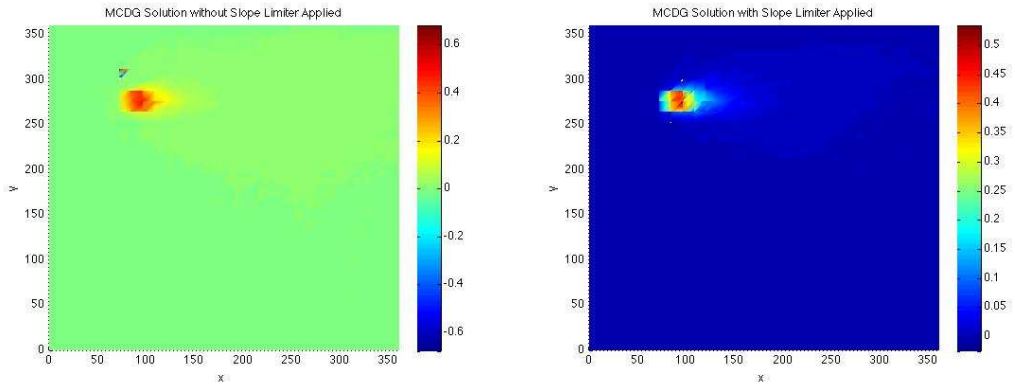


Figure 4.7 : Porous medium of Tarbert formation: MCDG solution without slope limiter applied (left), and MCDG solution with slope limiter applied (right). For both solutions, $T = 1.0, N_t = 2, c_{upper} = 1.1, c_{lower} = 0.1$, the diffusion coefficient is $K_b = 2 \times 10^{-4}$.

fields used to generate random permeability fields in my thesis is extremely large, from $\mathcal{O}(10^{-8})$ to $\mathcal{O}(10^4)$ mD. In order to address this difficulty, I use upper c_{upper} and lower c_{lower} bounds to limit the average solution on each element in $[c_{lower}, c_{upper}]$. If the maximum of average solution on all elements $\max_{E\in\mathcal{E}_h} \bar{c}_E$ is larger than c_{upper}, or the minimum of average solution on all elements $min_{E\in\mathcal{E}_h} \bar{c}_E$ is less than c_{lower}, then this simulation is not added into the last step in the MCDG algorithm. Therefore, the criteria for a qualified simulation is:

$$\max_{E\in\mathcal{E}_h} \bar{c}_E \leq c_{upper} \quad \text{and} \quad \min_{E\in\mathcal{E}_h} \bar{c}_E \geq c_{lower} \tag{4.18}$$

where $\mathcal{E}_h$ is the subdivision (or mesh), E is the element, $\bar{c}_E$ is the average solution of c_h on element $E : \bar{c}_E = \frac{1}{|E|} \int_E c_h$.

Table 4.1 indicates that as the upper bound increases and the lower bound decreases, the number of disqualified simulations decreases. Additionally, for fixed upper bound and lower bound, number of qualified simulations decreases as the length of simulation time, T, increases.

Table 4.1 indicates that the proportion of qualified solutions in all 10000 simula-

Table 4.1 : Number of disqualified simulations with different upper and lower bound in 10000 samples. The size of permeability field is 15×15. The size of the mesh is 30×30.

Tarbert formation, $\Delta t = 1/6$	$T = 0.5$	$T = 1.0$	$T = 1.5$
$c_{upper} = 5.0, c_{lower} = -2.0$	14	58	216
$c_{upper} = 2.0, c_{lower} = -1.0$	57	117	280
$c_{upper} = 1.5, c_{lower} = -0.5$	107	187	362
$c_{upper} = 1.4, c_{lower} = -0.4$	136	216	401
$c_{upper} = 1.3, c_{lower} = -0.3$	182	277	492
$c_{upper} = 1.2, c_{lower} = -0.2$	297	457	706
$c_{upper} = 1.1, c_{lower} = -0.1$	873	1159	1372

tions are more than 97% when c_{upper} is equal to 1.2, c_{lower} is equal to -0.2 and T is equal to 0.5. Therefore, I conclude that combining slope limiter with constraint (4.18) and applying lower resolution permeability to finer meshes can address the overshoot and undershoot phenomena in a large number of simulations effectively. Fig 4.8 gives the MCDG solution with c_{upper} is equal to 1.1, c_{lower} is equal to -0.1. The slope limiter is applied after each time step. Piecewise linear approximations are used, with parameters $\sigma = 1, \epsilon = 1$.

When the upper and lower bounds are relaxed to 5.0 and -2.0, MCDG solutions in Fig 4.9 are similar to the ones in Fig 4.8. It validates that the technique of combining slope limiter, constraints (4.18) and application of lower resolution permeability to finer meshes is stable and convincing. Rare overshoot or undershoot phenomena occurs in MCDG solutions with the relaxed upper and lower bounds.

Solutions in Fig 4.8 and Fig 4.9 are obtained by using the permeability realizations

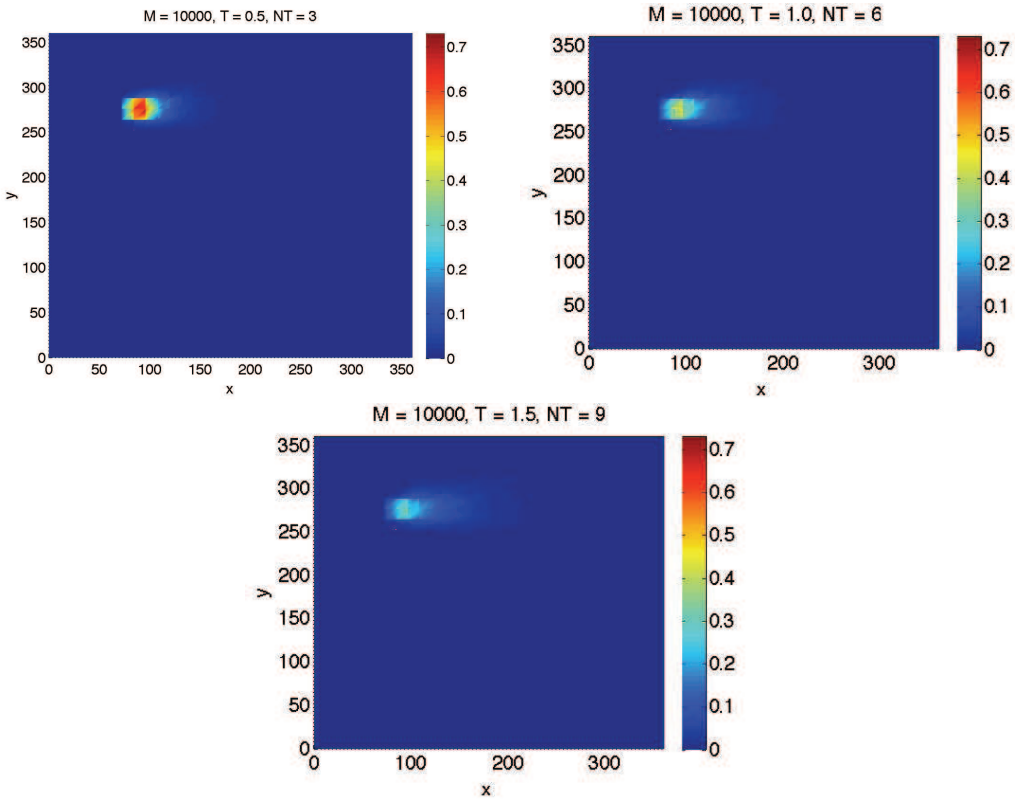


Figure 4.8 : Porous Medium of Tarbert formation: MCDG solution to the transport problem with boundary conditions (4.14) at $T = 0.5$ (left in first row), $T = 1.0$ (right in the first row), $T = 1.5$ (second row) with 10000 simulations. Time step $\Delta t = 1/6$, $c_{upper} = 1.1, c_{lower} = -0.1$. The slope limiter is applied after each time step. The diffusion coefficient is 2×10^{-4}. The size of the permeability field is 15×15. The size of the mesh is 30×30.

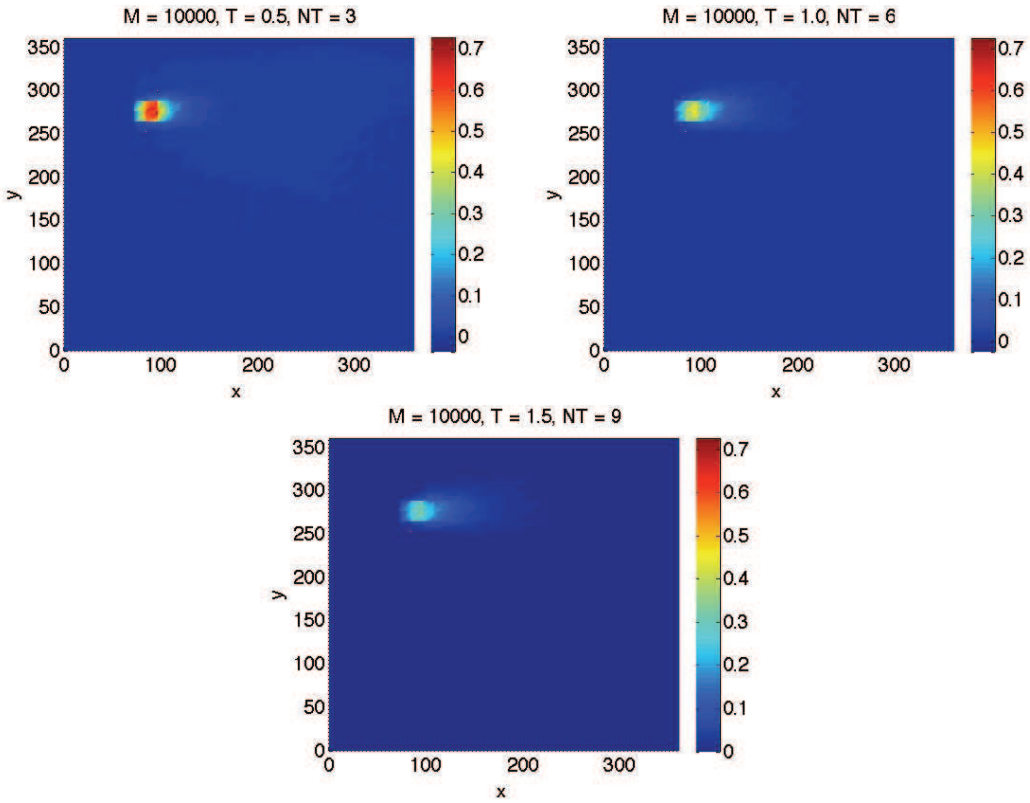


Figure 4.9 : Porous Medium of Tarbert formation: MCDG solution to the transport problem with boundary conditions (4.14) at $T = 0.5$ (left in first row), $T = 1.0$ (right in the first row), $T = 1.5$ (second row) with 10000 simulations. Time step $\Delta t = 1/6$, $c_{upper} = 5.0, c_{lower} = -2.0$. The slope limiter is applied after each time step. The diffusion coefficient is 2×10^{-4}. The size of the permeability field is 15×15. The size of the mesh is 30×30.

Table 4.2 : Number of disqualified simulations with different upper and lower bound in 10000 samples. The size of permeability realizations is 30×30. The size of the mesh is 30×30.

Tarbert formation, $\Delta t = 1/6$	$T = 0.5$	$T = 1.0$	$T = 1.5$
$c_{upper} = 5.0, c_{lower} = -2.0$	47	140	356
$c_{upper} = 2.0, c_{lower} = -1.0$	182	303	517
$c_{upper} = 1.5, c_{lower} = -0.5$	428	646	929
$c_{upper} = 1.4, c_{lower} = -0.4$	586	810	1095
$c_{upper} = 1.3, c_{lower} = -0.3$	854	1083	1369
$c_{upper} = 1.2, c_{lower} = -0.2$	1431	1691	1888
$c_{upper} = 1.1, c_{lower} = -0.1$	2857	2930	3085

of size 15×15 on the mesh of size 30×30. In order to gain the conclusion that applying lower resolution permeability to finer mesh can avoid more overshoot and undershoot solutions to the transport equation, I compare the results obtained by 30×30 permeability with the ones by 15×15 permeability. The size of the mesh is 30×30 for both examples. I use second order kernel PCA to generate 10000 realizations of permeability field of size 30×30. The kernel matrix is build by using the permeabilities of the porous Medium of Tarbert formation in SPE10 model (as described in section 3.3.1), which is the same as the one generating the 15×15 permeability fields.

Table 4.2 shows the number of disqualified solutions obtained by applying 10000 30×30 permeability realizations onto a 30×30 mesh.

In order to compare the difference between Table 4.1 and Table 4.2, Fig 4.10 compare the number of disqualified solutions in these two examples. Dash circle

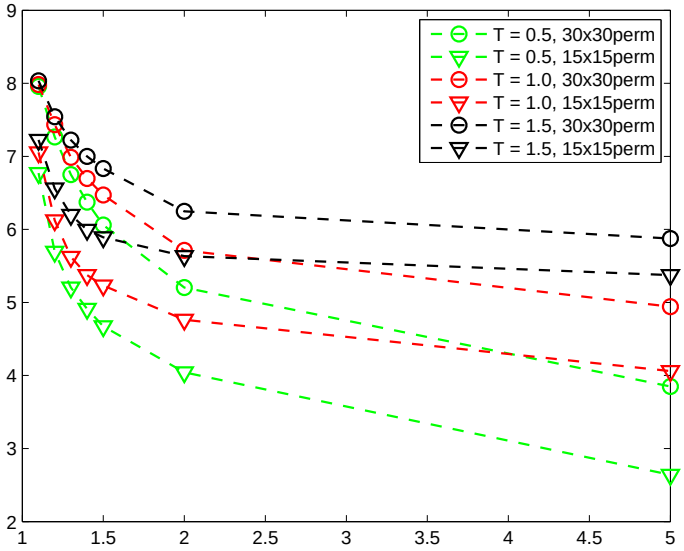


Figure 4.10 : Porous medium of Tarbert formation: Number of disqualified solutions by using 15×15 permeability fields (dash downward-pointing triangle line) and 30×30 permeability fields (dash circle line). X-axis represents c_{upper} and Y-axis represent $\log(N_{dq})$, where N_{dq} denotes the number of disqualified solutions. The mesh is the same for both cases and its size is 30×30.

line represents the result by applying 30×30 permeability onto 30×30 mesh. Dash downward-pointing triangle line represents the result by applying 15×15 permeability onto 30×30 mesh. Each color represents a certain time length. It shows that for a given color (same T), the dash circle line is always above the dash downward-pointing triangle line. That means more qualified solutions are obtained by using 15×15 permeability than by 30×30 permeability on the same 30×30 mesh.

Fig 4.11 interprets the differences between these two cases. Fig 4.11 indicates that the averages velocity fields of two cases are similar. However, the average velocity fields generated by 30×30 permeability fields are more oscillatory than the ones generated by 15×15 permeability fields. Therefore, more overshoot and undershoot solutions are generated in this case. I conclude that the more oscillation there exists

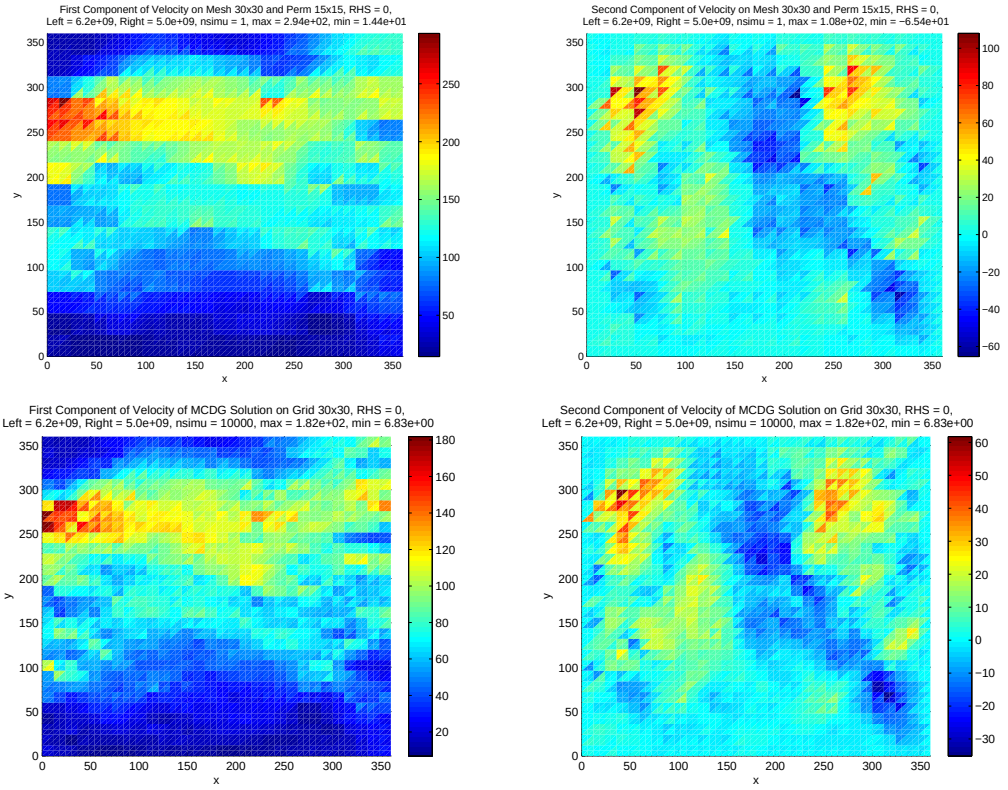


Figure 4.11 : Porous medium of Tarbert formation: First component (first column) and second component (second column) of average velocity fields generated by 15×15 (first row) and 30×30 (second row) permeability fields on the same mesh of size 30×30.

in the velocity fields, the more disqualified solutions are obtained.

Additionally, the maximum of the first and second component of the average velocity field generated by 15×15 permeability fields are larger than the ones generated by 30×30 permeability fields. To investigate the reason, recall

$$\boldsymbol{\beta} = -K_a \nabla u. \tag{4.19}$$

where K_a is defined in (4.13). Therefore, the difference can be validated by comparing the average of the 10000 permeability realizations generated by Kernel PCA for these two cases.

Diffusion coefficient $\tilde{K}_b$ depends on velocity

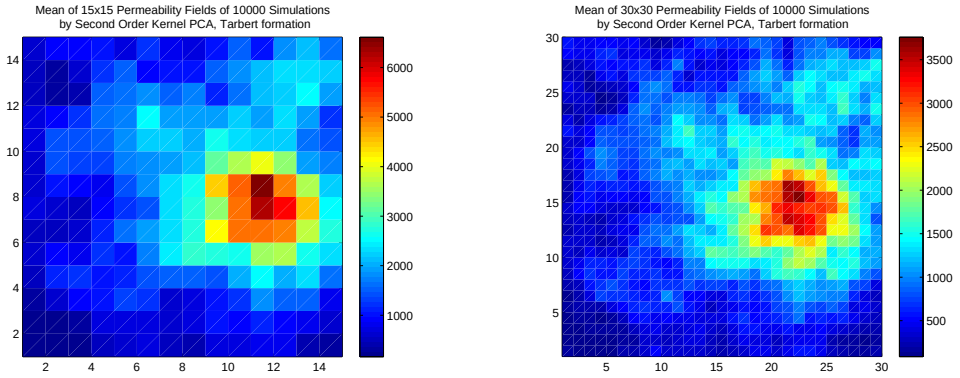


Figure 4.12 : Porous medium of Tarbert formation: Average of 10000 permeability realizations of size 15×15 (left) and 30×30 (right). Both of them are generated by using second order kernel PCA. In two cases, the kernel matrix is build by the permeability values of Porous medium of Tarbert formation in SPE10 model as described in Section 3.3.1.

In reality, the coefficient $\tilde{K}_b(u)$ depends on fluid velocity and is called diffusion-dispersion matrix coefficient. In the rest of this section, I discuss the same numerical example as above, but take $\tilde{K}_b(u)$ as

$$\tilde{K}_b(u) = K_m \mathbf{I} + \frac{\alpha_l}{|u|} \begin{bmatrix} u_x^2 & u_x u_y \\ u_x u_y & u_y^2 \end{bmatrix} + \frac{\alpha_t}{|u|} \begin{bmatrix} u_x^2 & -u_x u_y \\ -u_x u_y & u_y^2 \end{bmatrix} \tag{4.20}$$

where

- u_x: the first component of $-K_a \nabla u$,
- u_y: the second component of $-K_a \nabla u$,
- $|u| = \sqrt{u_x^2 + u_y^2}$,
- K_m is the molecular diffusion coefficient and is taken to be 2.0×10^{-4} in this numerical example,
- α_l: longitudinal dispersivity,

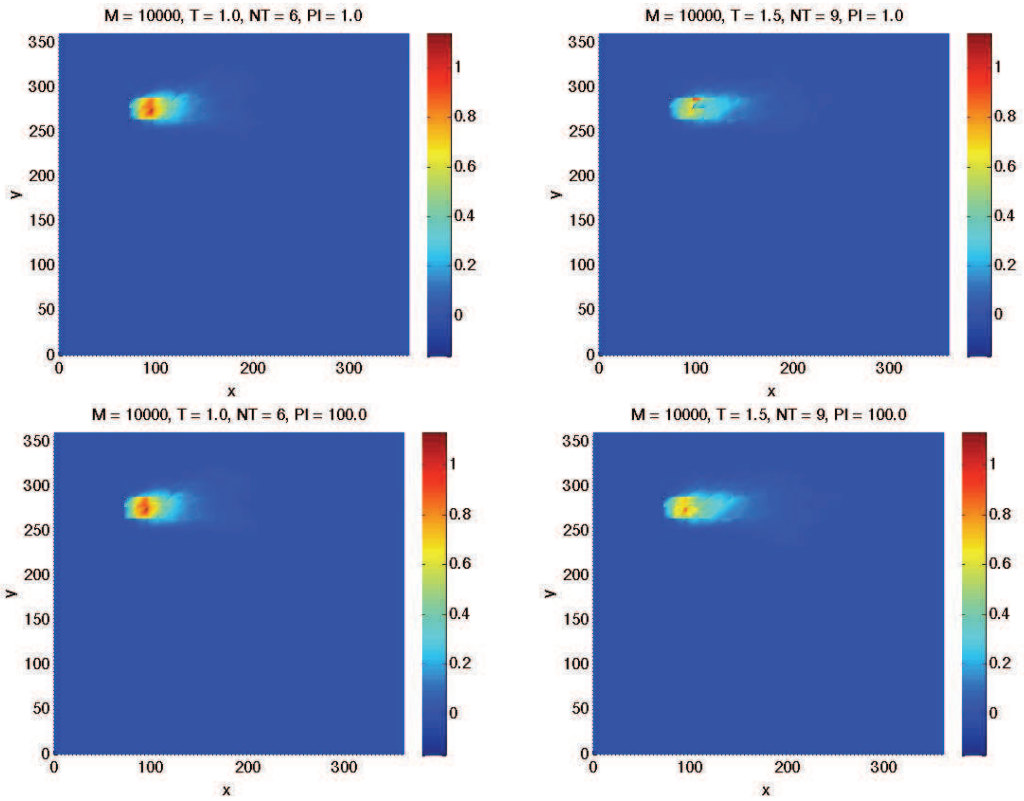


Figure 4.13 : First row: Peclet number is equal to 1.0 and the MCDG solutions when $T = 1$ (left) and $T = 1.5$ (right); Second row: Peclet number is equal to 100.0 and the MCDG solutions when $T = 1$ (left) and $T = 1.5$ (right).

- α_t: transverse dispersivity.

Additionally, we have $\alpha_l = \frac{L}{Pl}$, where L is the length of the domain and Pl is the Peclet number, and $\alpha_t = \alpha_l/10$. In the numerical example, we take $L = 360m, Pl = 1, 100$. The MCDG solutions for $Pl = 1$ and $Pl = 100$ are shown in Fig 4.13.

Fig 4.13 shows the similar results as the one when $\tilde{K}_b$ is a constant that the contaminant spreads toward the entire domain and does not leave the original location as time elapses. Furthermore, we observe the diffusion phenomena as we increase the time.

4.4.4 Porous Medium of Upper Ness Formation

In section 3.3.4, the second order kernel PCA is used to generate random permeability fields for two cases: Porous medium of Tarbert formation and porous medium of Upper Ness formation. Fig 3.2 implies that more complex geological structures, channels, appear in the porous medium of Upper Ness formation, while the geological structure of porous medium of Tarbert formation is relatively simple. In the previous section, I discuss the approach in section 4.3 to approximate the solution to the coupled flow and transport equation for porous medium of Tarbert formation. In this section, I apply this approach to porous medium of Upper Ness formation.

The initial solution in this case is:

$$c_0 = \begin{cases} 1.0 & (x, y) \in [24, 36] \times [180, 192], \\ 0 & \text{elsewhere.} \end{cases}$$

which is shown in Fig 4.14.

In last section, I conclude that we need to apply lower resolution permeability onto a finer mesh in order to obtain a more accurate approximation of the solution to the coupled problem. Since the geological structure of porous medium of Upper Ness formation is more complex than porous medium of Tarbert formation, I apply 15×15 permeability field onto 60×60 mesh, which is finer than the one of porous medium of Tarbert formation. The permeability realizations are generated by second order kernel PCA. The average of 10000 permeability realizations for porous medium of Upper Ness formation and the 60×60 mesh are given in Fig 4.15.

By applying these 10000 permeability realizations to solve flow equation with boundary conditions (4.15), I obtain 10000 velocity realizations of size 60×60. Fig 4.16 shows the average of the first and second component of velocity field for porous medium of Upper Ness formation.

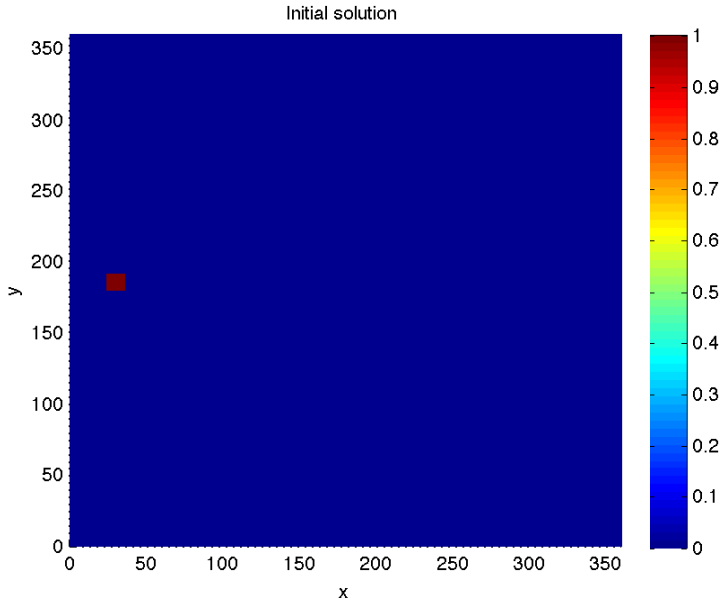


Figure 4.14 : Porous medium of Upper Ness formation: initial solution, where $D_I = \{(x, y)|(x, y) \in [24, 36] \times [180, 192]\}$.

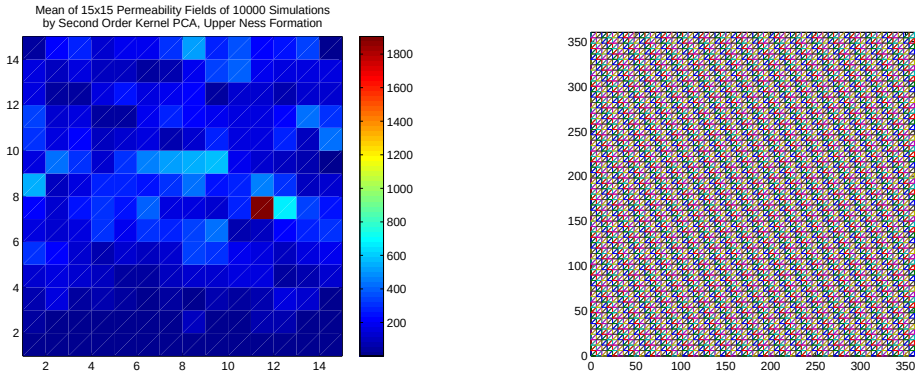


Figure 4.15 : Porous medium of Upper Ness formation: average of 10000 15×15 permeability realizations, which are generated by second order kernel PCA (left) and the mesh for pressure and flow equations (right) for porous medium of Upper Ness formation.

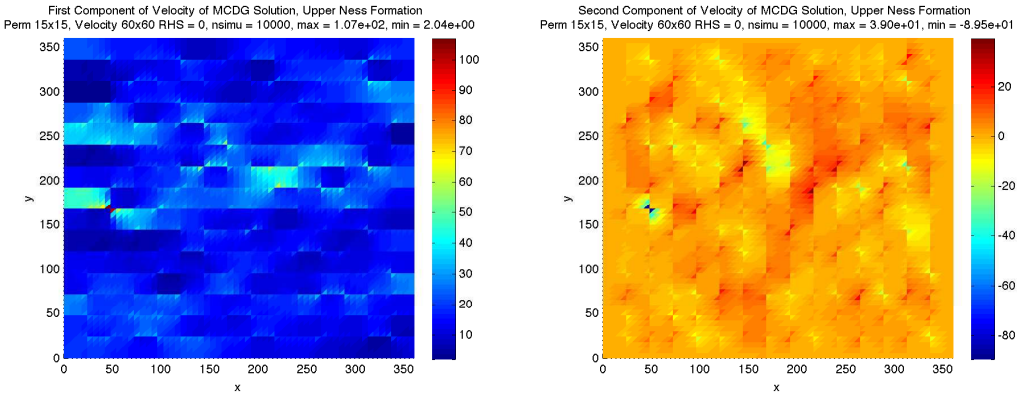


Figure 4.16 : Porous medium of Upper Ness formation: average of first and second component of flow velocity of 10000 realizations. They are generated from pressure equation with boundary conditions (4.14) by using DG method.

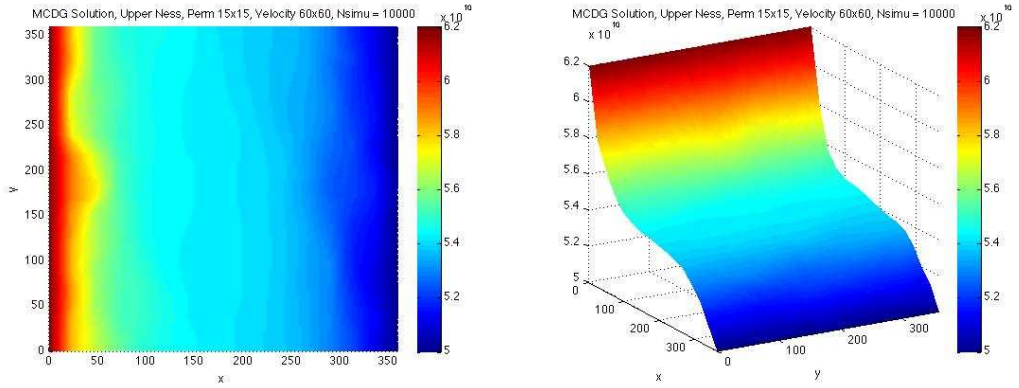


Figure 4.17 : Porous medium of Upper Ness formation: MCDG solution to the flow equation with boundary conditions (4.15) with 10000 simulations in different views. The size of permeability field is 15×15. The size of mesh is 60×60

Fig 4.16 implies that the average of first component of velocity is positive in the whole domain and the second component is positive in most of the domain. It indicates that the direction of transport of concentration is from left to right. This is validated by the boundary conditions (4.15), where the pressure on the left boundary is larger than the pressure on the right boundary. Fig 4.17 shows the MCDG solution to the flow equation with boundary conditions (4.15) with 10000 simulations.

Solving Transport Problem with MCDG

In this section, I apply the 10000 flow velocity realizations generated from pressure equation by MCDG, individually, onto transport equation with boundary conditions (4.15). It generates 10000 solutions to the transport equation. Because channel structures appear in the porous medium of Upper Ness formation, the oscillation of permeability is larger than for the porous medium of Tarbert formation. Therefore, more uncertainty and oscillation take place in the 10000 velocity realizations. It causes more overshoot and undershoot solutions in this case. I apply slope limiter and the lower and upper bounds constraints to address this difficulty.

Fig 4.18 shows the MCDG solution to the transport equation with boundary conditions (4.15) at different time T. The slope limiter is applied after each time step. Upper bound c_{uppper} is equal to 1.1 and lower bound c_{lower} is equal to -0.1. The time interval Δt is equal to 1/6.

Fig 4.18 shows the transport of concentration of the contaminant at $T = 1.0, 1.5, 2.0, 3.0$. As we observe, the contaminant does not leave the initial location completely as time elapses, while part of the contaminant spreads toward the entire domain and along the direction from right to left.

Since channel structure, a more complex geological structure, appears in the porous medium of Upper Ness formation, more overshoot and undershoot solutions are excepted to appear in this case. Table 4.3 shows the number of disqualified solutions to this case.

By comparing Table 4.3 with Table 4.1, we conclude that it is an effective technique to limit solutions to the transport equation with high level of uncertainty and oscillation in velocity field by applying permeability fields of lower resolution onto finer meshes. Even if more complex geological structure appears, this technique can

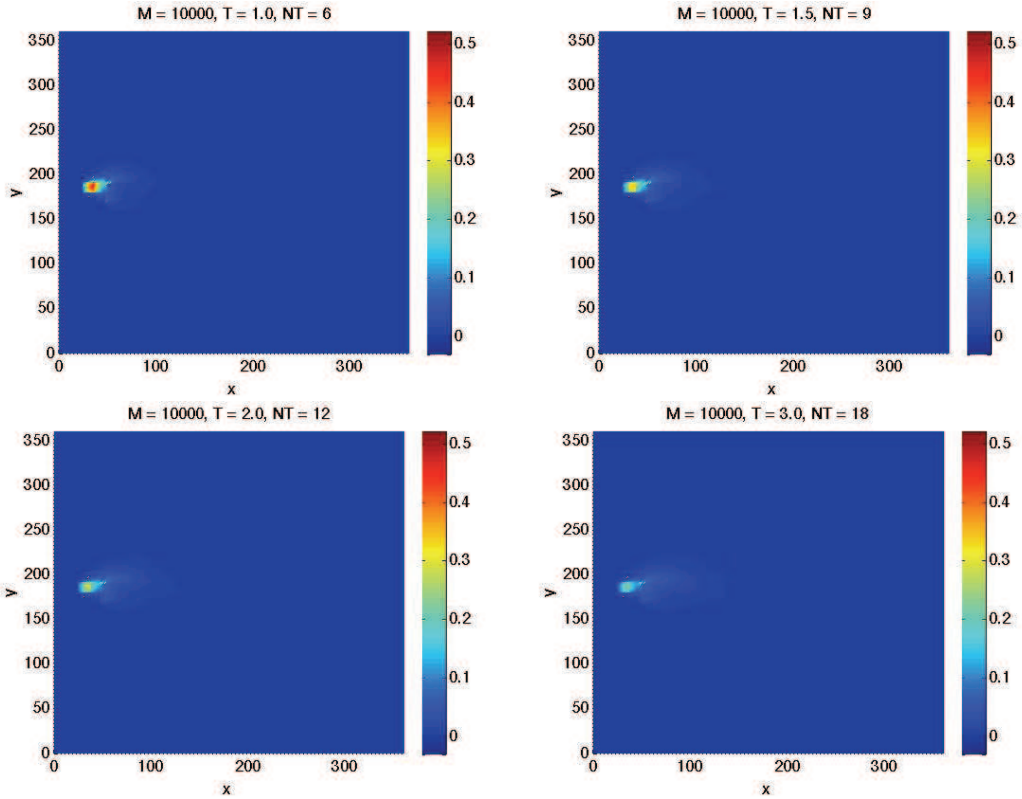


Figure 4.18 : Porous medium of Upper Ness formation: MCDG solution to the transport problem with boundary conditions (4.15) at $T = 1.0$ (left in first row), $T = 1.5$ (right in the first row), $T = 2.0$ (left in the second row), $T = 3.0$ (right in the second row) with 10000 simulations. $c_{upper} = 1.1, c_{lower} = -0.1$. The size of permeability fields is 15×15. The size of the mesh is 60×60. Second order kernel PCA is used to generated the 10000 permeability realizations.

Table 4.3 : Number of disqualified solutions in 10000 simulations to the transport equation with boundary conditions 4.15 with various upper and lower bounds. The size of permeability is 15×15. The size of mesh is 60×60.

Upper Ness formation, $\Delta t = 1/6$	$T = 1.0$	$T = 1.5$	$T = 2.0$	$T = 3.0$
$c_{upper} = 5.0, c_{lower} = -2.0$	70	149	277	628
$c_{upper} = 2.0, c_{lower} = -1.0$	246	287	421	756
$c_{upper} = 1.5, c_{lower} = -0.5$	497	531	623	934
$c_{upper} = 1.4, c_{lower} = -0.4$	621	633	722	1023
$c_{upper} = 1.3, c_{lower} = -0.3$	790	804	882	1165
$c_{upper} = 1.2, c_{lower} = -0.2$	1097	1064	1135	1378
$c_{upper} = 1.1, c_{lower} = -0.1$	1867	1754	1757	1914

reduce the number of the overshoot and undershoot solutions significantly. But it cannot remove all of them, because DG methods do not satisfy maximum principle. For the porous medium of Upper Ness formation, around 90% of solutions are in the range $[-0.2, 1.2]$ for various time T. It achieves the goal of addressing overshoot and undershoot solutions when more complex geological structure appears.

To conclude, I apply the approach proposed in section 4.3 for the porous media of Tarbert formation and Upper Ness formation. Using this approach, the solution to this couple system shows that part of the contaminant remains in the initial location and the plume of the contaminant moves to the left-hand side gradually as time elapses. The results obtained by this approach are realistic, because it takes the uncertainty of velocity field into consideration.

4.4.5 Solvers

For the porous medium of Upper Ness formation, the size of the grid is 60×60, which means it contains $60 \times 60 \times 2 = 7200$ triangular elements. The basis functions in DG method for the transport problem are polynomials of degree one on each element. Therefore, there are $60 \times 60 \times 2 \times 3 = 21600$ degrees of freedom in this case. I apply PETSc [55–57] toolkit on Rice parallel machine STIC to solve the linear system $Ax = b$. Twenty Intel Core i7 processors in STIC are used to solve 10000 simulations. It takes about 160 minutes to finish 10000 simulations when $T = 3.0$ and $N_t = 18$. In order to show the advantage of PETSc, I compare the computing time of PETSc and LAPACK on solving transport equation with boundary conditions (4.15) once, where $T = 1.0$ and $N_t = 6$. The size of mesh is 30×30. It takes 0.53 seconds by PETSc and 14.51 seconds by LAPACK. The computation is more than 20 times faster by PETSc. In PETSc, the solver I use is GMRES with preconditioner ILU. GMRES takes 14 iterations to reach the relative tolerance 10^{-12} and the absolute tolerance 10^{-50}. In LAPACK, I use dgesv routine which implements complete LU decompsition. The machine I use is caampc52 with one Intel Core i3 processor. Furthermore, since the global matrix A is very sparse, PETSc stores the matrix in a sparse format, which saves a large portion of memory resources. For example, for porous medium of Upper Ness formation, the size of mesh is 60×60. If we use the data type of double precision, it costs

$$\frac{21600 \times 21600 \times 8}{1024 \times 1024 \times 1024} = 3.8GB$$

memory to store the matrix in the dense format. It means it is impossible to accomplish one time simulation on STIC by dgesv routine, because the memory of each processor in STIC is only $1.5GB$. Therefore, PETSc plays a significant role in

the computation for the case of porous medium of Upper Ness formation. It saves memory resources and improves the speed of computation.

Chapter 5

Conclusions

Uncertainty quantification in complex systems governed by partial differential equations with random input data has attracted tremendous research attention in recent years. In my thesis, I focus on the numerical solution of coupled elliptic and parabolic partial differential equations with random coefficients. In the coupled system, the uncertainty of convection term in transport equation is caused by random permeability field in flow equation. Knowledge of permeability is significant in determining the flow characteristics of hydrocarbons in oil and gas reservoir, and of ground water in aquifers. Therefore, my research is practically important to oil industry and water pollution management.

In my thesis, I first establish the Monte Carlo discontinuous Galerkin (MCDG) method for solving parabolic partial differential equations with random coefficients. This equation models the transport of a chemical species through a porous medium. Monte Carlo method is used to randomly sample the coefficients generated from a certain probability distribution in the outcome space. Discontinuous Galerkin (DG) method is used to discretize in the space domain with backward Euler discretization in the time domain. The one-dimensional numerical example shows that for a given number of simulations, MCDG converges when the number of intervals in space domain increases. Additionally, as expected, when the number of intervals in space domain is known, MCDG converges when the number of samples increases. In the two-dimensional example, I develop one approach to approximate the solution. The

numerical results are obtained by MCDG method with random convection term. I conclude that when the level of uncertainty in convection term is high, slope limiter is needed to fix the overshoot and undershoot solutions. Applying lower resolution permeability on finer mesh is also effective to reduce the overshoot and undershoot phenomena.

Secondly, I apply the polynomial kernel Principal Component Analysis (PCA) of second order to generate random permeability fields based on the permeability data in the SPE10 model. Two types of porous media are discussed. In the Upper Ness type, more complex geological structures, such as channels, appear, while in the Tarbert type, no channel structure exists. Kernel PCA is an effective method to generate random fields, because it only requires the eigen-decomposition of the kernel matrix generated by a small set of realizations in geological simulations and is able to capture the complex geological structures, such as channels. In my thesis, I apply kernel PCA to generate realizations of random permeability fields for these two types of porous media. The large variances of the realizations in both types of the porous media imply the apparent differences of these realizations, thus, suggest the high uncertainty among them. Then, I apply these realizations to solve a single phase flow model with random permeability fields by using MCDG. The size of the grid where random permeability fields are generated is finer than the sizes of the grids for single phase flow model. For the porous media of both types, the computing time on the coarser grids is significantly reduced, compared with the CPU time on finer meshes. The computation is performed on STIC, a powerful parallel machine at Rice University with multiprocessors.

In Chapter 4, I study a coupled system, which consists of flow equation and transport equation. The system models the transport of contaminant in a saturated porous

medium. In the flow equation, the coefficient represents a random permeability field. The randomness of the permeability field leads the convection term (velocity) in the transport equation to be a random field. I develop one approach to solve this model problem. In this approach, I use MCDG to generate M realizations of the convection term. Then, I apply each of them individually to solve the transport equation and obtain M solutions to the coupled system. By averaging the M solutions, I obtain the MCDG solution to this coupled system. In the numerical examples, the solutions obtained by this approach show that only part of the contaminant leaves the initial location and the rest of the contaminant still stays in the initial location but diffuses as time elapses and spreads toward the entire domain in the direction determined by the realizations of the convection term. I conclude that the solution by the this approach models the transport of the concentration of contaminant more accurately, because it takes the uncertainty in the convection term into consideration. Due to the high level of uncertainty in the realizations of the convection term, I use slope limiter and lower and upper bound constraints to correct the overshoot and undershoot solutions. Because DG methods do not satisfy the maximum principle, the positivity of the solution is not guaranteed even the slope limiter is applied. However, I conclude that in order to avoid more overshoot and undershoot solutions, we need to apply permeabilities of lower resolution on finer meshes. This idea works effectively to reduce certain amount of solutions which are not positive. Through this technique, among the simulations in MCDG, most of the solutions to the transport problem remain bounded for the porous medium of Tarbert formation as well as for the porous medium of Upper Ness formation.

Additionally, I discuss the case when the diffusion coefficient in the transport equation depends on fluid velocity for the Tarbert format porous medium. In that

situation, the coefficient is a matrix and is called diffusion-dispersion matrix coefficient. The similar numerical results are obtained as the numerical examples where the diffusion coefficient is constant. The results show that the contaminant moves toward the entire domain and does not leave the original location completely. Meanwhile, as times goes by, the contaminant diffuses obviously.

Furthermore, for the porous medium of Upper Ness formation, I use PETSc to solve the coupled system and the computing time is significantly reduced compared with the time by LAPACK.

The future work of this dissertation includes extending this coupled model into a fully coupled one, which is known as miscible displacement problem. The viscosity in my thesis is constant. In the fully coupled system, the viscosity in the flow equation depends on the concentration of contaminant computed from the transport equation, and the convection term in the transport equation depends on the pressure computed from the flow equation. I plan to apply MCDG to solve the miscible displacement problem.

Finally, I plan to extend my current work to a three-dimensional case.

Bibliography

[1] K. Liu, "Discontinuous Galerkin Methods for Elliptic Partial Differential Equations with Random Coefficients: Department of Computational and Applied Mathematics, Rice University, Houston, Texas, USA,"

[2] S. Ghori, J. Heller, and A. Singh, "An efficient model for computing plume concentration in Aquifers," *Environmetrics*, vol. 3(2), pp. 149–167, 1992.

[3] Y. Efendiev and W. Luo, "Preconditioning Markov chain Monte Carlo simulations using coarse-scale models," *SIAM Journal on Scientific Computing*, vol. 28, pp. 776–803, 2006.

[4] J. Proft and B. Rivière, "Discontinuous Galerkin methods for convection-diffusion equations for varying and vanishing diffusivity," *International Journal of Numerical Analysis and Modelling*, vol. 6(4), pp. 533–561, 2009.

[5] P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equation*. Springer-Verlag, New York, 1992.

[6] S. Mishra and C. Schwab, "Sparse tensor multi-level monte carlo finite volume methods for hyperbolic conservation laws with random initial data," tech. rep., Seminar fur Angewandte Mathamatik, Eidgenossische Technische Hochschule, 2010.

[7] M. Kleiber and T. D. Hien, *The Stochastic Finite Element Method*. Wiley, 1992.

[8] I. Babuška, R. Tempone, and G. E. Zouraris, "Galerkin finite element approximations of stochastic elliptic partial dierential equations," *SIAM Journal on Numerical Analysis*, vol. 42(2), pp. 800–825, 2004.

[9] W. Powell, *Approximate dynamic programming: solving the curses of dimensionality.* John Wiley and Son, 2007.

[10] S. Smolyak, "Quadrature and interpolation formulas for tensor products of certain classes of functions," *Doklady Akademii Nauk*, vol. 4, pp. 240–243, 1963.

[11] F. Nobile, R. Tempone, and C. Webster, "A sparse grid stochastic collocation method for partial differential equations with random input data," *SIAM Journal on Numerical Analysis*, vol. 46, pp. 2309–2345, 2008.

[12] F. Nobile, R. Tempone, and C. Webster, "An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data," *SIAM Journal on Numerical Analysis*, vol. 46, pp. 2411–2442, 2008.

[13] X. Ma and N. Zabaras, "An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations," *Journal of Computational Physics*, vol. 228(8), pp. 3084–3113, 2009.

[14] G. Fishman, *Monte Carlo: Concepts, Algorithms, Applications.* Springer-Verlag, 1996.

[15] H. Anderson, "Metropolis, Monte Carlo and the MANIAC," *Los Alamos Science*, vol. 14, pp. 96–108, 1986.

[16] S. Mark and M. S., *Applications of Monte Carlo Method in Science and Engineering.* InTech, 2011.

[17] I. Babuška, R. Tempone, and G. E. Zouraris, "Solving elliptic boundary value problems with uncertain coecients by the finite element method: the stochastic formulation," *Computer Methods in Applied Mechanics and Engineering*, vol. 194(12-16), pp. 1251–1294, 2005.

[18] S. Sun and J. Liu, "A locally conservative finite element method based on piecewise constraint enrichment of the continuous Galerkin method," *SIAM Journal on Scientific Computing*, vol. 31, pp. 2528–2548, 2009.

[19] W. Reed and T. Hill, "Triangular mesh methods for the neutron transport equation," tech. rep., Los Alamos Scientific Laboratory, 1973.

[20] M. Wheeler, "An elliptic collocation-finite element method with interior penalties," *SIAM Journal on Numerical Analysis*, vol. 15, pp. 152–161, 1978.

[21] D. Arnold, "An interior penalty finite element method with discontinuous elements," *SIAM Journal on Numerical Analysis*, vol. 19, pp. 742–760, 1982.

[22] B. Rivière, M. Wheeler, and V. Girault, "Improved energy estimates for interior penalty, constrained and discontinuous Galerkin methods for elliptic problems," *Part I, Computational Geosciences*, vol. 3, pp. 337–360, 1999.

[23] C. Dawson, S. Sun, and M. Wheeler, "Compatible algorithms for coupled flow and transport," *Computer Methods in Applied Mechanics and Engineering*, vol. 193, pp. 2565–2580, 2004.

[24] B. Ganis, I. Yotov, and M. Zhong, "A stochastic mortar mixed finite element method for flow in porous media with multiple rock types," tech. rep., University of Pittsburgh, 2010.

[25] S. Sun and M. Wheeler, "Discontinuous Galerkin methods for coupled flow and reactive transport flow," *Applied Numerical Mathematics*, vol. 52, pp. 273–298, 2005.

[26] Y. Epshteyn and B. Rivière, "Convergence of high order methods for miscible displacement," *International Journal of Numerical Analysis and Modeling*, vol. 5, pp. 47–63, 2008.

[27] B. Rivière, *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations : Theory and Implementation.* Society for Industrial and Applied Mathematics, 2008.

[28] S. Zein, V. Rath, and C. Clauser, "A multidimensional Markov chain model for simulating stochastic permeability conditioned by pressure measures," *The International Journal of Multiphysics*, vol. 4(4), pp. 359–374, 2010.

[29] J. Chilés and P. Delfiner, *Geostatistics: modeling spatial uncertainty.* Wiley, 1999.

[30] W. Li, "A fixed-path Markov Chain algorithm for conditional simulation of discrete spatial variables," *Mathematical Geology*, vol. 39(2), pp. 321–335, 2007.

[31] B. Scholkopf, A. Smola, and K.-R. Muller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

[32] X. Ma and N. Zabaras, "Kernel principal component analysis for stochastic input model generation," *Journal of Computational Physics*, vol. 230, pp. 7311–7331, 2011.

[33] M. Christie and M. Blunt, "Tenth SPE Comparative Solution Project: A Comparison of Upscaling Techniques," *SPE Reservoir Engineering and Evaluation*, vol. 4, pp. 308–317, 2001.

[34] P. D. Lax, *Functional Analysis*. Pure and Applied Mathematics: A Wiley-Interscience Series of Texts, Monographs, and Traces, 2002.

[35] M. Loève, *Probability Theory. I, fourth edition, in: Graduate Texts in Mathematics*. Springer-Verlag, New York, 1977.

[36] B. Øksendal, *Stochastic Differential Equations. An Introduction with Applications*. Springer-Verlag, 1998.

[37] C. Geuzaine and J.-F. Remacle, "Gmsh: A Three-dimensional Finite Element Mesh Generator with Built-in Pre- and Post-processing Facilities," *International Journal for Numerical Methods in Engineering*, vol. 79(11), pp. 1309–1331, 2009.

[38] L. Gelhar and C. Axness, "Three-dimensional stochastic analysis of macrodispersion in aquifers," *Water Resources Research*, vol. 19, pp. 161–180, 1983.

[39] L. Gelhar, *Stochastic subsurface hydrology*. Englewood Cliffs, New Jersey: Prentice Hall, 1992.

[40] G. Dagan, *Flow and transport in porous formation*. Springer-Verlag. Berlin, Heidelberg, 1989.

[41] D. Kolyukhin and M. Espedal, "Numerical calculation of effective permeability by double randomization Monte Carlo Method," *International Journal of Numerical Analysis and Modeling*, vol. 7(4), pp. 607–618, 2010.

[42] D. Fenton and D. Griffiths, "Flow through earth dams with spatially random permeability," *Geomechanics Research Center, Colorado School of Mines*, 1994.

[43] K. Mela and J. Louie, "Correlation length and fractal dimension interpretation from seismic data using variograms and power spectra," *Geophysics*, vol. 66, pp. 1372–1378, 2001.

[44] Y. Yortsos and N. Afaleg, "The permeability variogram from pressure transients of multiple wells: theory and 1-D application," *Society of Petroleum Engineers Journal*, vol. 2(3), 1997.

[45] R. Hinkelman, *Efficient numerical methods and information-processing techniques for modeling hydro- and environmental systems*. Springer, 2005.

[46] D. Zhang and Z. Lu, "An efficient, high-order perturbation approach for flow in random porous media via Karhunen-Loéve and polynomial expansions," *Journal of Computational Physics*, vol. 194, pp. 773–794, 2004.

[47] P. Sarma, L. Durlofsky, and K. Aziz, "Kernel Principal Component Analysis for Efficient, Differentiable Parameterization of Multipoint Geostatistics," *Mathematical Geology*, vol. 40, pp. 3–32, 2008.

[48] J. Kwok and I. Tsang, "The pre-image problem with Kernel methods," *IEEE Trans Neural Network*, vol. 15, pp. 1517–1525, 2004.

[49] H. Hoteit, R. M. Ackerer, J. Erhel, and B. Philipe, "New two-dimensional slope limiters for discontinuous Galerkin methods on arbitrary meshes," *International Journal for Numerical Methods in Engineering*, vol. 61, pp. 2566–2593, 2004.

[50] L. Durlofsky, B. Engquist, and S. Osher, "Triangle based adaptive stencils for the solution of hyperbolic conservation laws," *Journal of Computational Physics*, vol. 98, pp. 64–73, 1992.

[51] B. Cockburn and C.-W. Shu, "The Range-Kutta discontinuous Galerkin method for conservation laws V," *Journal of Computational Physics*, vol. 141, pp. 199–224, 1998.

[52] H. Darcy, *Les Fontaines Publiques de la Ville de Dijon*. 1856.

[53] E. Jones and C. Lineweaver, "Pressure-temperature Phase Diagram of the Earth," *ASP Conference Series*, vol. 430, pp. 145–151, 2010.

[54] M. Holz, S. Heil, and A. Sacco, "Temperature-dependent Self-diffusion Coefficients of Water and Six Selected Molecular Liquids for Calibration in Accurate 1H NMR PFG Measurements," *Physical Chemistry Chemical Physics*, vol. 2, pp. 4740–4742, 2000.

[55] S. Balay, J. Brown, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, "PETSc Web page," 2012. http://www.mcs.anl.gov/petsc.

[56] S. Balay, J. Brown, , K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, "PETSc users manual," Tech. Rep. ANL-95/11 - Revision 3.3, Argonne National Laboratory, 2012.

[57] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, "Efficient management of parallelism in object oriented numerical software libraries," in *Modern Software Tools in Scientific Computing* (E. Arge, A. M. Bruaset, and H. P. Langtangen, eds.), pp. 163–202, Birkhäuser Press, 1997.